

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

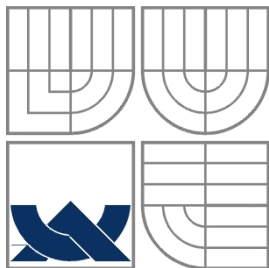
## TRANSCEIVER A TNC PRO DATOVOU KOMUNIKACI NA UHF S OBVODEM CC1020

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

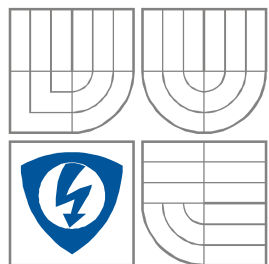
AUTOR PRÁCE  
AUTHOR

PETR HLAVICA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

# TRANSCEIVER A TNC PRO DATOVOU KOMUNIKACI NA UHF S OBVODEM CC1020

TRANSCEIVER AND TNC FOR DATA COMMUNICATION IN UHF BAND WITH CC1020 CHIP

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Petr Hlavica

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. Tomáš Urbanec, Ph.D.

BRNO, 2008

---

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Petr Hlavica  
Bytem: Kelč 345, Kelč, 756 43  
Narozen/a (datum a místo): 5. prosince 1982 ve Valašském Meziříčí

(dále jen „autor“)

a

### 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií  
se sídlem Údolní 53, Brno, 602 00  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:  
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika  
(dále jen „nabyvatel“)

### Čl. 1

#### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
- ☒ diplomová práce
- ☐ bakalářská práce
- ☐ jiná práce, jejíž druh je specifikován jako .....  
(dále jen VŠKP nebo dílo)

Název VŠKP: Transceiver a TNC pro datovou komunikaci na UHF s obvodem CC1020  
Vedoucí/ školitel VŠKP: Ing. Tomáš Urbanec, Ph.D.  
Ústav: Ústav radioelektroniky  
Datum obhajoby VŠKP: \_\_\_\_\_

VŠKP odevzdal autor nabyvateli\*:

- ☒ v tištěné formě – počet exemplářů: 2
- ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

---

\* hodící se zaškrtněte

---

## Článek 2

### Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☒ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3

### Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 30. května 2008

.....  
Nabyvatel

.....  
Autor

---

## **Abstrakt**

Cílem diplomové práce Transceiver a TNC pro datovou komunikaci na UHF s obvodem CC1020 je realizace hardwaru a softwaru pro mikrokontrolér řady Atmel AVR umožňující přenos dat rádiovou cestou. Jedná se o experiment, zda je obvod CC1020 využitelný pro realizaci TNC. Práce se zabývá studiem struktury protokolů AX25 a KISS, vlastnostmi integrovaného RF transceiveru CC1020 firmy Chipcon a návrhem PA a LNA zesilovačů pro zvýšení dosahu systému.

## **Klíčová slova**

TNC, Terminal Node Controller, CC1020, Chipcon, AX25, KISS, paket, Atmel, AVR, PA, LNA, vysílač, přijímač, scrambling, bit stuffing

---

## **Abstract**

The aim of Master's thesis Transceiver and TNC for Data Communication in UHF Band with CC1020 Chip is design of unit, which provides a data transfer by the packet radio net. It is an experiment, whether CC1020 chip is possible to use for TNC design. The thesis consists of study AX25 a KISS protocols, study of CC1020 features, design of PA and LNA and programming control software for Atmel AVR microcontroller.

## **Key words**

TNC, Terminal Node Controller, CC1020, Chipcon, AX25, KISS, packet, Atmel, AVR, PA, LNA, transmitter, receiver, scrambling, bit stuffing

---

## Bibliografická citace

HLAVICA, P. *Transceiver a TNC pro datovou komunikaci na UHF s obvodem CC1020*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 45 s, 3 příl. Vedoucí diplomové práce Ing. Tomáš Urbanec, Ph.D.

---

## Prohlášení

Prohlašuji, že svou diplomovou práci na téma Transceiver a TNC pro datovou komunikaci na UHF s obvodem CC1020 jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 30. května 2008

.....  
podpis autora

## Poděkování

Děkuji vedoucímu diplomové práce Ing. Tomáši Urbancovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 30. května 2008

.....  
podpis autora



# Obsah

1	Úvod .....	1
2	Přenos dat .....	3
2.1	AX25 protokol, rámec protokolu AX25 .....	3
2.2	TNC v KISS módu, datový rámec .....	5
2.2.1	Typy rámců .....	6
2.3	Převod dat mezi rámci protokolů AX25 a KISS .....	7
3	CC1020 .....	8
3.1	Vlastnosti .....	8
3.1.1	Základní parametry .....	8
3.1.2	Vnitřní struktura obvodu .....	9
3.1.3	Doporučené zapojení, pouzdro a popis pinů .....	9
3.2	Řídící a datové rozhraní obvodu .....	13
3.3	Řídící registry .....	15
3.4	Módy přenosu dat .....	17
3.5	Rychlost přenosu dat .....	17
3.6	Příjem dat .....	18
3.7	Vysílání dat .....	20
3.8	Kódování dat .....	21
4	Obvodové řešení modulu .....	22
4.1	UHF zesilovače .....	22
4.1.1	PA stupeň vysílače .....	22
4.1.2	LNA stupeň přijímače .....	24
4.1.3	Filtr vyšších harmonických složek .....	26
4.2	Zapojení modulu .....	27
4.3	Rozhraní modulu .....	29
5	Programové řešení .....	30
5.1	Terminálový program .....	30
5.2	Modul cc1020.c .....	30
5.2.1	Konfigurační pole DefaultConfig433 .....	30
5.2.2	Funkce read_cc1020 .....	31
5.2.3	Funkce write_cc1020 .....	31
5.2.4	Funkce setup_cc1020 .....	32
5.2.5	Funkce reset_cc1020 .....	32
5.2.6	Funkce calibrate_cc1020 .....	32
5.2.7	Funkce setup_cc1020_pd .....	32
5.2.8	Funkce wakeup_cc1020_rx a wakeup_cc1020_tx .....	32
5.2.9	Funkce setup_cc1020_rx .....	33
5.2.10	Funkce setup_cc1020_tx .....	33
5.3	Modul tnc.c .....	33
5.3.1	Hlavní funkce main .....	33
5.3.2	Funkce Check_TNC .....	36
5.3.3	Funkce Get_Random .....	37
5.3.4	Funkce FCS_TX .....	37

5.3.5	Funkce FCS_RX .....	37
5.3.6	Funkce setup_pd.....	38
5.3.7	Funkce Transmit_data .....	38
5.3.8	Funkce Receive_data.....	38
5.3.9	Obsluha vnějšího přerušení RF_data_clock .....	38
5.3.10	Funkce RF_data_encode .....	39
5.3.11	Funkce RF_data_decode .....	39
5.3.12	Funkce Send_Kiss_To_PC.....	40
5.3.13	Funkce Send_Byte_To_UART .....	40
5.3.14	Obsluha přerušení příjmu ze sériové linky Receive_Kiss_From_PC .....	40
5.3.15	Obsluha přerušení časovače Timer1_over(void).....	40
6	TNC CC1020.....	41
6.1	Parametry TNC .....	41
6.2	Návod k obsluze .....	41
7	Závěr.....	44
8	Seznam použité literatury .....	45
9	Přílohy .....	46
9.1	Deska plošného spoje .....	46
9.2	Seznam součástek.....	47
9.3	Instalace a nastavení terminálového programu .....	48

## Seznam obrázků

Obr. 1	Schéma scrambleru.....	5
Obr. 2	Vztah mezi rámcem KISS a AX25 .....	7
Obr. 3	Blokové schéma obvodu CC1020 .....	9
Obr. 4	Doporučené zapojení obvodu .....	10
Obr. 5	Pouzdro obvodu CC1020, pohled shora.....	11
Obr. 6	Rozhraní pro mikrokontrolér.....	13
Obr. 7	Časování řídicího rozhraní, zápis do registru.....	13
Obr. 8	Časování řídicího rozhraní, čtení z registru.....	14
Obr. 9	Časování datového rozhraní v NRZ módu.....	14
Obr. 10	Časování datového rozhraní v Manchester módu .....	15
Obr. 11	Blokové schéma demodulátoru .....	19
Obr. 12	Výkon a spotřeba CC1020 vs. nastavení registru PA_POWER .....	21
Obr. 13	Principiální schéma zesilovačů .....	22
Obr. 14	Model PA .....	23
Obr. 15	Přizpůsobení PA pro pracovní kmitočet .....	23
Obr. 16	Přizpůsobení PA pro 0 - 2,5 GHz .....	24
Obr. 17	Model LNA.....	25
Obr. 18	Přizpůsobení LNA pro pracovní kmitočet.....	25
Obr. 19	Přizpůsobení LNA pro 0 - 2,5 GHz.....	26
Obr. 20	Filtr vyšších harmonických složek.....	26
Obr. 21	Charakteristiky filtru vyšších harmonických .....	27
Obr. 22	Schéma zapojení modulu .....	28

<i>Obr. 23 Rozhraní modulu.....</i>	<i>29</i>
<i>Obr. 24 Princip SPI.....</i>	<i>31</i>
<i>Obr. 25 Princip funkce TNC .....</i>	<i>35</i>
<i>Obr. 26 Schéma generátoru náhodného čísla.....</i>	<i>37</i>
<i>Obr. 27 Okno zprávy terminálového programu AGW .....</i>	<i>42</i>
<i>Obr. 28 Typické hodnoty RSSI pro pásmo 433 MHz a různé šířky pásma kanálu.....</i>	<i>43</i>
<i>Obr. 29 Osazení modulu ze strany součástek a popis pinů.....</i>	<i>46</i>
<i>Obr. 30 Osazení modulu ze strany spojů.....</i>	<i>46</i>
<i>Obr. 31 Konfigurace rádiového kanálu - TNC Setup.....</i>	<i>48</i>
<i>Obr. 32 Konfigurace rádiového kanálu - TNC Commands .....</i>	<i>49</i>
<i>Obr. 33 Terminálový program AGWterm .....</i>	<i>49</i>

## Seznam tabulek

<i>Tab. 1 Citlivost obvodu CC1020 v závislosti na datové rychlosti, zdvihu a módu přenosu .....</i>	<i>1</i>
<i>Tab. 2 U a S rámec (příkazové rámce).....</i>	<i>3</i>
<i>Tab. 3 I rámec (informační rámec).....</i>	<i>3</i>
<i>Tab. 4 Rámec KISS protokolu .....</i>	<i>5</i>
<i>Tab. 5 Speciální znaky KISS módu.....</i>	<i>6</i>
<i>Tab. 6 Typy rámců KISS protokolu .....</i>	<i>6</i>
<i>Tab. 7 Základní parametry RF transceiveru.....</i>	<i>8</i>
<i>Tab. 8 Doporučené pracovní podmínky .....</i>	<i>8</i>
<i>Tab. 9 Přehled součástek doporučeného zapojení obvodu .....</i>	<i>10</i>
<i>Tab. 10 Popis pinů obvodu CC1020 .....</i>	<i>12</i>
<i>Tab. 11 Přehled registrů obvodu CC1020 .....</i>	<i>16</i>
<i>Tab. 12 Módy přenosu dat, vlastnosti .....</i>	<i>17</i>
<i>Tab. 13 Překódovací tabulka DIV1.....</i>	<i>18</i>
<i>Tab. 14 Překódovací tabulka DIV2.....</i>	<i>18</i>
<i>Tab. 15 Registr CLOCK_X.....</i>	<i>18</i>
<i>Tab. 16 Překódovací tabulka ADC_DIV.....</i>	<i>19</i>
<i>Tab. 17 Ukázka práce s buffery, pakety a ukazovátky.....</i>	<i>36</i>
<i>Tab. 18 Seznam konfiguračních příkazů TNC.....</i>	<i>42</i>

# 1 Úvod

Cílem Diplomové práce je realizace datového transceiveru pro packet rádio s obvodem CC1020 fy. Chipcon. Datový přenos bude zprostředkován terminálovým řídicím uzlem TNC (Terminal Node Controller), který bude s hostujícím osobním počítačem komunikovat v KISS módu.

Jedná se o experiment, zda výše jmenovaný obvod lze použít pro konstrukci takového zařízení pro pásmo 433 MHz. Cílem je konstrukce TNC maximální jednoduchosti kompatibilní se standardem 9k6 G3RUH, jehož popis lze nalézt v [11]. Navrhovaný TNC má zpracovávat data na úrovni fyzické a spojové vrstvy. Data na úrovni ostatních vrstev zpracovává speciální terminálový program běžící v připojeném osobním počítači. Vývoj terminálového programu není součástí práce, bude použit již existující, volně šířitelný software.

Tento projekt navazuje na [1] a [2], kde již bylo prakticky odzkoušeno vysílání a příjem dat. Obvod CC1010 se však ukázal jako nevhodný pro příjem dat při požadované kmitočtové modulaci s nízkým kmitočtovým zdvihem. Proto byl pro realizaci tohoto projektu zvolen obvod CC1020 stejného výrobce. Schopnosti příjmu signálu při nízkých kmitočtových zdvizech ukazuje tab. 1. Z uvedených parametrů se lze domnívat, že je možné dosáhnout spolehlivého příjmu dat při zdvihu signálu  $\pm 2750$  Hz a rychlosti přenosu 9600 Bd. Tento obvod je pouze RF transceiverem, proto je nutné ho doplnit vhodným mikrokontrolérem s řídicím programem.

Tab. 1 Citlivost obvodu CC1020 v závislosti na datové rychlosti, zdvihu a módu přenosu

Datová rychlost [kBd]	Zdvih [kHz]	Citlivost [dBm]		
		NRZ	Manchester	UART
2,4	$\pm 2,025$	-115	-118	-115
4,8	$\pm 2,475$	-112	-112	-112
9,6	$\pm 4,95$	-110	-111	-110
19,2	$\pm 9,9$	-107	-108	-107
38,4	$\pm 19,8$	-104	-104	-104
76,8	$\pm 36$	-101	-101	-101
153,6	$\pm 72$	-96	-97	-96

V případě realizovatelnosti TNC se zadaným obvodem mají být implementovány funkce pro přeladění pracovního kmitočtu, změnu výkonu vysílače a jako doplněk funkce pro změnu zdvihu a změnu rychlosti přenosu dat.

Z výše uvedeného vyplývá, že je potřeba provést rozvahu nad zajištěním dostupnosti požadovaných funkcí. Jednou z variant je použití již existujícího terminálového programu s možností definice vlastních řídicích signálů. Další variantou, a také pravděpodobnější, je ovládat funkce TNC pomocí datových zpráv speciálního formátu, u kterého je nepravděpodobný výskyt v reálném provozu. Tyto zprávy se nebudou vysílat.

Praktické využití této práce je v oblasti sdělovací techniky, pro amatérské účely. Jedna z předností tohoto přenosu je cenová dostupnost a provozní náklady, které v dnešní době

představují jedny z nejsledovanějších podmínek provozu. Proto je tento způsob provozu, tak populární mezi amatéry.

V dalším popisu uvádím pouze informace mající přímý vztah k této práci a jsou postačující k pochopení předkládaného řešení. Nepodstatné informace jako vlastnosti obvodu CC1020 v pásmu 804 – 940 Mhz nebo podrobnou strukturu protokolu AX25 neuvádím, lze je nalézt v [3], [4] [5] a [6]. Text je psán tak, aby byl vystižen vztah jednotlivých částí práce. Např., aby bylo zřejmé, zda je možné obvodem vysílat platný paket nebo zda je možné obvodem paket přijímat.

## 2 Přenos dat

V TNC dochází ke dvěma přenosům dat – galvanickým spojením mezi osobním počítačem a TNC a mezi dvěma TNC rádiovou cestou. TNC má za úkol data přijatá z PC transformovat do formátu pro rádiovou cestu a odeslat a naopak.

### 2.1 AX25 protokol, rámec protokolu AX25

Pro přenos dat rádiovou cestou se u všech typů modulací používá protokol AX25. Protokol vychází z protokolu X25 určeného pro počítačové sítě. Základní jednotkou přenosu je jeden bit, proto hovoříme o tzv. bitově orientovaném protokolu. Pro lepší přehlednost však bity seskupujeme po osmi do tzv. oktětů neboli bajtů. Několik za sebou jdoucích správně organizovaných bajtů tvoří tzv. rámec (angl. frame) a několik rámců za sebou pak tzv. paket. Rámce lze rozdělit na řídicí a informační, příklady nejběžnějších rámců ukazují tab. 2 a tab. 3. U všech polí se jako první přenáší nejméně významný bit, výjimku tvoří pole FCS u něhož je první vysílán nejvíce významný bit. Data se před vstupem do modulátoru vysílače převedou do kódu NRZI, kde změnou stavu se kóduje nulový bit, zatímco při jedničkovém bitu zůstává stav nezměněn. Podrobnější popis najdeme v [5] nebo v [6].

Tab. 2 U a S rámec (příkazové rámce)

Fend	Address	Control	FCS	Fend
01111110	112/560 bitů	8 bitů	N*8bitů	01111110

Tab. 3 I rámec (informační rámec)

Fend	Address	Control	PID	Info	FCS	Fend
01111110	112/560 bitů	8 bitů	8 bitů	N*8bitů	16 bitů	01111110

Fend - hlavička

Začátek i konec rámce je ohraničen hlavičkou, koncová hlavička může být začátkem nového rámce. Tvoří ji posloupnost 01111110, tato posloupnost se nikde jinde v rámci nesmí vyskytovat. Pro lepší synchronizaci přijímače se v praxi vysílá několik hlaviček za sebou, obvykle po dobu několika set milisekund.

Aby se uvnitř rámce nemohla sekvence návěští objevit, je při vysílání po pěti jedničkách vsunuta nula bez ohledu na následující hodnotu a při příjmu je vypuštěna. Tato procedura se nazývá bit stuffing.

### Address - adresové pole

Adresové pole obsahuje adresu příjemce, odesílatele, případně nódů (síťový uzel, obdoba převaděče). Adresa se skládá ze šesti ASCII znaků posunutých o 1 bit doleva, viz [5], [6], a znaku SSID (Secondary Station Identification). Adresové pole obsahuje vždy sudá čísla, pouze u SSID posledního adresového pole je přičtena jednička a SSID je tak liché. Tím je jednoznačně odděleno adresové pole od dalších částí paketu.

### Control - řídící pole

Tento bajt označuje druh rámce, např. I rámec (informační), RR rámec (příjem připraven).

### PID – identifikace protokolu

Pole identifikace protokolu se vyskytuje jen u rámců, které nesou informaci a rozlišuje spojení uživatel-nód a nód-nód.

### Info – datové pole

Obsahuje přenášená data, maximální délka pole je 256 bajtů. Informace se přenáší jako ASCII znaky.

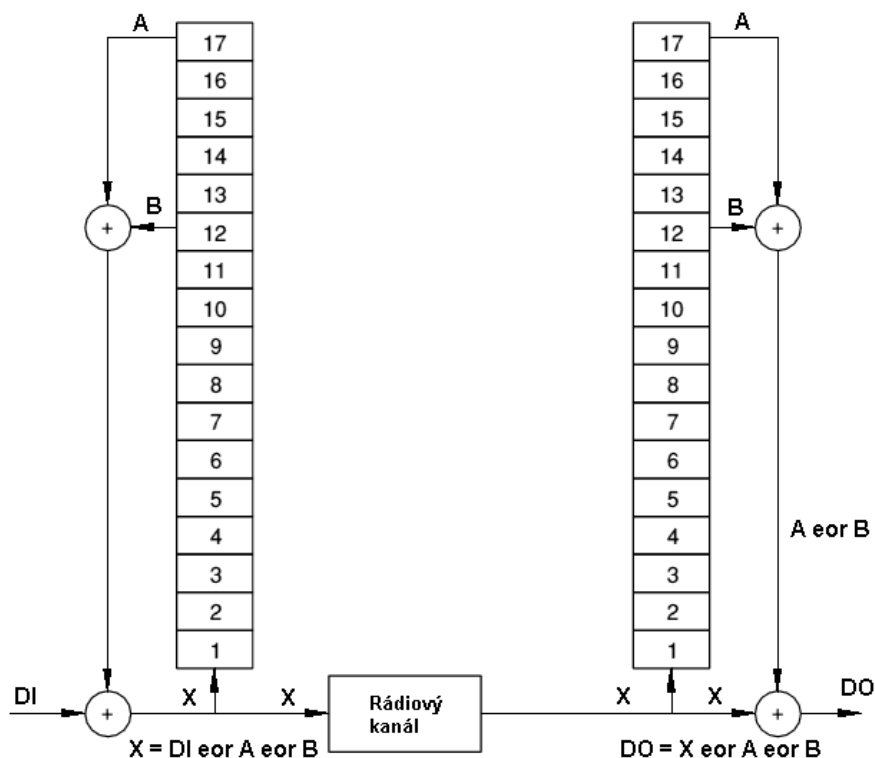
### FCS – cyklické zabezpečení

Toto dvoubajtové číslo je výstupem cyklického kódu, počítá se ze všech přenášených bajtů mimo hlavičky rámce. Cyklické zabezpečení je při příjmu znovu vypočítáno a pokud hodnoty nesouhlasí, je v rámci chyba. Vysílá se jedničkový doplněk výsledku cyklického zabezpečení. Generační polynom má tvar::

$$g(x) = x^{16} + x^{12} + x^5 + 1 \quad (1)$$

Ve standardu G3RUH se data scramblují (kódují pseudonáhodnou posloupností), čímž je dosaženo rovnoměrnějšího využití spektra a lepší bitové synchronizace demodulátoru. Princip scrambleru ukazuje obr.1, generační polynom má tvar:

$$g(x) = x^{17} + x^{12} + 1 \quad (2)$$



Obr. 1 Schéma scrambleru

## 2.2 TNC v KISS módu, datový rámec

KISS ("Keep It Simple, Stupid") protokol je jednoduchý rámcový přenos dat mezi osobním počítačem a TNC. Stavbu rámce ukazuje tab. 4. Datové pole se přebírá/převádí z/do rámce AX25. Rámce lze opět rozdělit na řídicí a informační. Podrobnější popis najdeme v [7] nebo v [8].

Tab. 4 Rámec KISS protokolu

Fend	Adr/Ctrl	Info	Fend
0xC0	1 bajt	1 - 256 bajtů	0xC0

Fend – hlavička

Začátek i konec rámce je ohraničen hlavičkou. Tvoří ji posloupnost 0xC0, tato posloupnost se nikde jinde v rámci nesmí vyskytovat.

Pokud se v rámci po úvodním znaku Fend vyskytuje další znak Fend není to chápáno jako prázdný rámec.

Znak Fend v datové části paketu je přeložen jako dvojice znaků Fesc a Tfend, při příjmu je tato kombinace přeložena zpět na znak Fend. Znak Fesc v datové části paketu je přeložen jako dvojice znaků Fesc a Tfesc, při příjmu je tato kombinace přeložena zpět na znak Fesc. Speciální znaky jsou v tab. 5.



Tab. 5 Speciální znaky KISS módu

Znak	Kód
Fend	0xC0
Fesc	0xDB
Tfend	0xDC
Tfesc	0xDD

Adr/Ctrl – adresové a řídicí pole

Obsahuje adresu portu TNC (bity 6:4) a informaci o jaký typ rámce se jedná (bity 3:0). Nejvýznamnější bit určuje zda přenos bude probíhat bez zabezpečení ochranným kódem, mód KISS, nebo se zabezpečením, mód SMACK.

Info – datové pole

Obsahuje přenášená data nebo parametry příkazu. Délka pole u datového rámce je maximálně 256 bajtů, datové pole řídicích rámců má mimo rámec *Set hardware* délku jeden bajt.

### 2.2.1 Typy rámců

Tab.6 ukazuje všechny typy rámců KISS protokolu. Rámce mají význam příkazů, řízení funkcí TNC. Prodlevy *TxDelay* a *TxTail* slouží k překonání přechodových dějů při klíčování vysílače. V praxi se po tyto doby opakovaně vysílá hlavička paketu protokolu AX25.

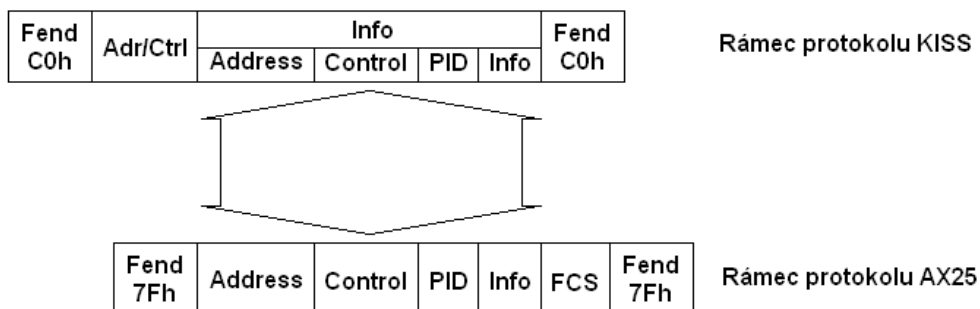
Před odvysíláním paketu se generuje náhodné číslo. Je-li toto číslo menší než parametr *Persistence*, TNC zaklíčuje a odešle všechny datové rámce ve frontě. Je-li generované číslo menší, TNC čeká po dobu *Slot time*. Poté opět generuje náhodné číslo, porovnává ho s *Persistence* a cyklus se opakuje dokud se nepodaří data odvysílat.

Tab. 6 Typy rámců KISS protokolu

Ctrl pole	Název	Obsah datového pole
x0h	Data frame	Data, která se budou vysílat, která byla přijata.
x1h	TxDelay	Doba prodlevy v 10 ms jednotkách po zaklíčování a před přenosem dat.
x2h	Persistence	Parametr přístupu k médiu
x3h	Slot time	Prokládací interval v 10ms jednotkách (10)
x4f	TxTail	Doba prodlevy v 10ms jednotkách přidržení vysílače po odvysílání dat.
x5h	Full duplex	0 je poloviční duplex, 1 je duplex (0)
x6h	Set hardware	Různé, podle druhu hardware (rychlost přenosu...)
FFh	Return	Ukončení KISS

## 2.3 Převod dat mezi rámcí protokolů AX25 a KISS

Vztah mezi rámcí obou protokolů ukazuje obr.2. Datová pole datových rámců KISS protokolu obsahují téměř kompletní rámec protokolu AX25. Úkolem TNC při vysílání je k těmto polím připojit hlavičky rámce Fend a cyklické zabezpečení FCS. Následně provést bit stuffing, převést data do kódu NRZI a provést scrambling. Takto připravený rámec je odvysílán. Při příjmu je postup obrácený.



Obr. 2 Vztah mezi rámcem KISS a AX25

### 3 CC1020

Obvod CC1020 je datový RF transceiver s jedním směšováním v pásmu ultra krátkých vln s integrovaným rozhraním pro připojení vnějšího mikrokontroléru s řídicím programem. Obvod je navržen tak, aby bylo potřeba připojit minimum vnějších součástek, převážně blokovacích kondenzátorů napájení.

#### 3.1 Vlastnosti

##### 3.1.1 Základní parametry

Tab. 7 Základní parametry RF transceiveru

Parametr	Min.	Typ.	Max.	Jednotka	Poznámka <sup>1)</sup>
Kmitočtový rozsah	402		940	MHz	Programovatelný v krocích po 300/600 Hz v pásmu 433/868 MHz
Datový tok	0,45		153,6	kBaud	
Kmitočtový zdvih modulační FSK	0		108/216	kHz	433/868 MHz
Výstupní výkon 433/868 MHz	-20		10/5	dBm	Do zátěže 50Ω, 433/868 MHz
Citlivost přijímače 433/868 MHz		-118		dBm	4,8 kBaud, NRZ, 2,475kHz kmitočtový zdvih, BER = 10 <sup>-3</sup>
Mf kmitočet		307,2		kHz	Vnitřní filtr
Odběr proudu TX 433/868 MHz		16,2/20,5		mA	Výstupní výkon 0 dBm
Odběr proudu RX		19,9		mA	
Modulace	OOK, FSK, GFSK				

Tab. 8 Doporučené pracovní podmínky

Parametr	Min.	Typ.	Max.	Jednotka	Poznámka <sup>2)</sup>
Napájecí napětí	2,7	3,3	3,6	V	
Rozsah teplot	-40		85	°C	
Pracovní kmitočet hlavního oscilátoru	4,9152	14,7456	19,6608	MHz	<sup>2)</sup>

1) Podrobnější popis parametrů obvodu nalezneme v [3].

2) Hlavní oscilátor slouží jako zdroj referenčního kmitočtu pro generátor rychlosti datového toku. Doporučené kmitočty jsou: 4,9152, 7,3728, 9,8304, 12,2880, 14,7456, 17,2032 a 19,6608 MHz, se kterými lze nastavit standardní rychlosti přenosu.

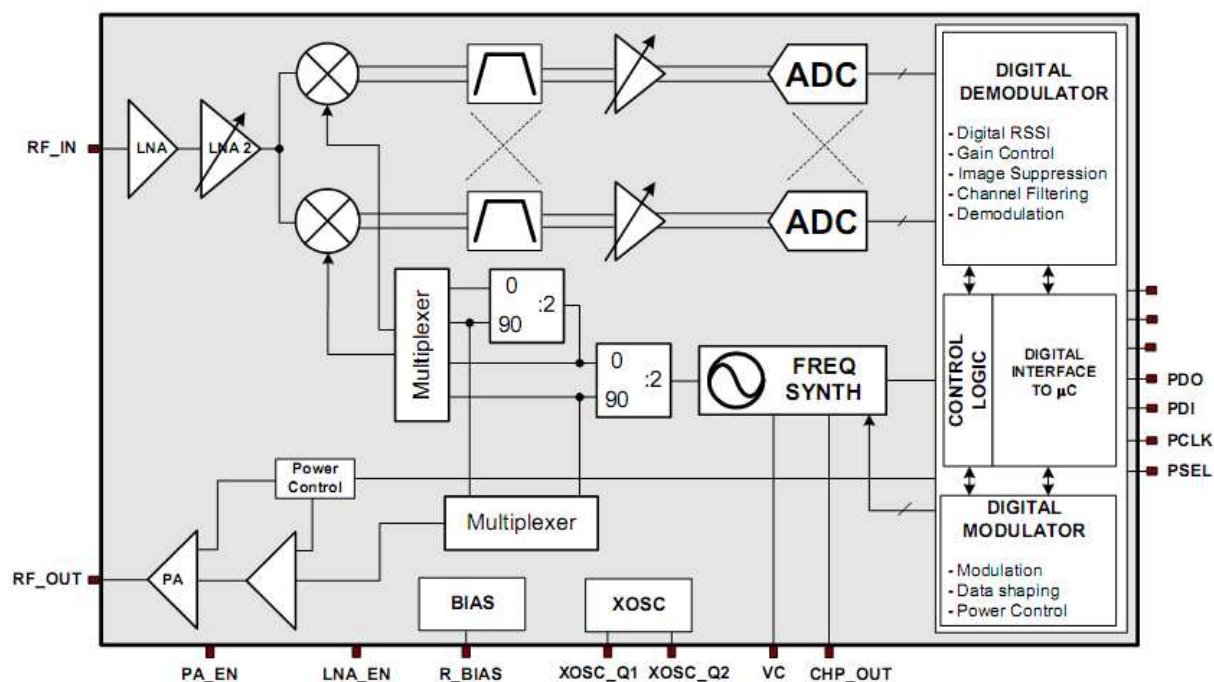
### 3.1.2 Vnitřní struktura obvodu

Obr. 3 ukazuje zjednodušené blokové schéma obvodu CC1020. Obvod je rozdělen na analogovou a digitální část. Tyto části mají oddělené napájení, napájecí piny nejsou na obrázku nakresleny.

Srdcem analogové části obvodu je kmitočtový syntezátor pracující v rozsahu kmitočtů 1,608 až 1,880 GHz. Syntezátor využívá vnějšího filtru smyčky fázového závěsu (PLL), který je připojen na signály VC a CHP\_OUT. Tím je umožněno měnit vlastnosti PLL podle potřeb aplikace. Za syntezátorem následují dvě děličky kmitočtu, každá s výstupem s fází 0° a 90° pro kvadraturní demodulátor. Výběr pásma se provádí multiplexery.

Přijímač je typu superheterodyn, přijímaný signál se zesílí vstupním nízkošumovým zesilovačem LNA a směšuje na mezifrekvenční kmitočet 307,2 kHz. Signál se dále filtruje, vzorkuje a navzorkovaný signál postupuje do demodulátoru v digitální části obvodu.

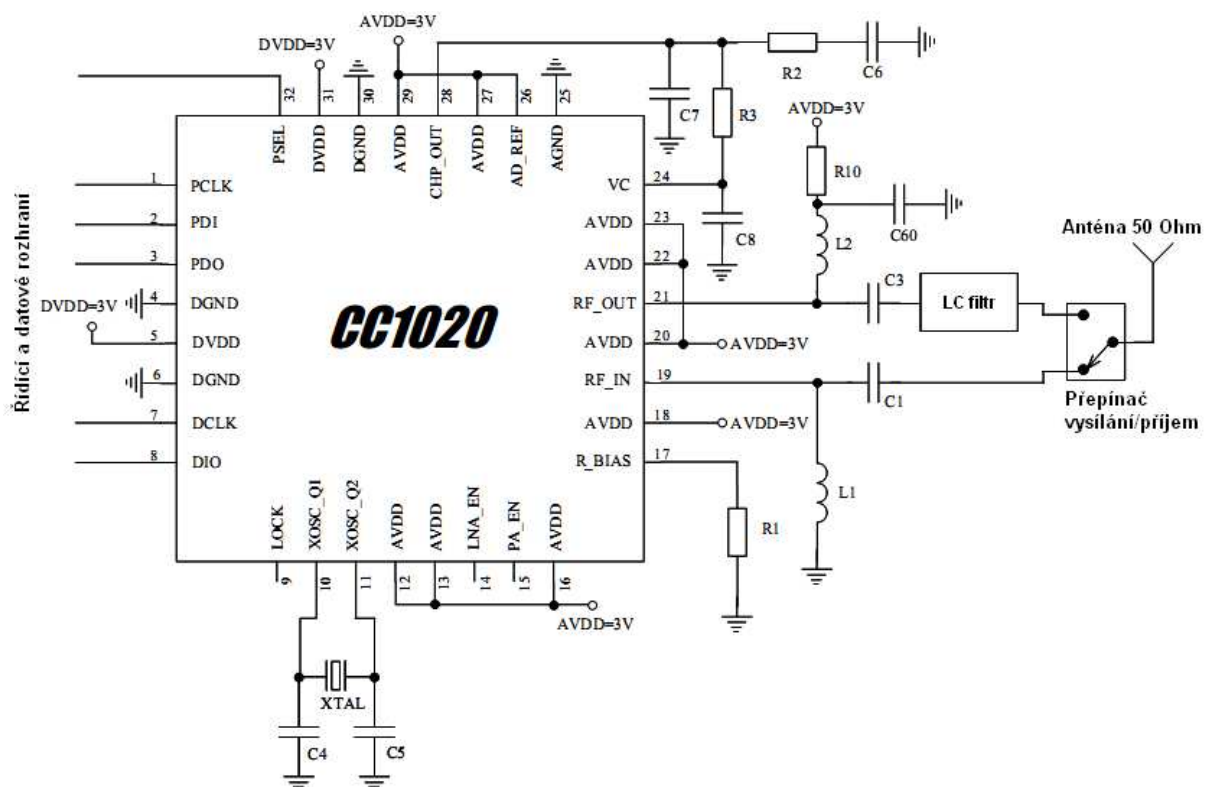
Při vysílání se do smyčky PLL zavádí modulační signál. Výstupní signál po výběru pásma multiplexerem postupuje do výstupního výkonového zesilovače PA.



Obr. 3 Blokové schéma obvodu CC1020

### 3.1.3 Doporučené zapojení, pouzdro a popis pinů

Doporučené zapojení obvodu ukazuje obr.4, z obrázku je i patrná schematická značka obvodu. Obvod ke své funkci vyžaduje minimální počet vnějších součástek. Ve schématu nejsou zakresleny blokovací kondenzátory napájecího rozvodu. Blokovací kondenzátor by měl být co nejblíže každého napájecího pinu, doporučená hodnota je přibližně 100pF. Popis součástek schématu je v tab. 9

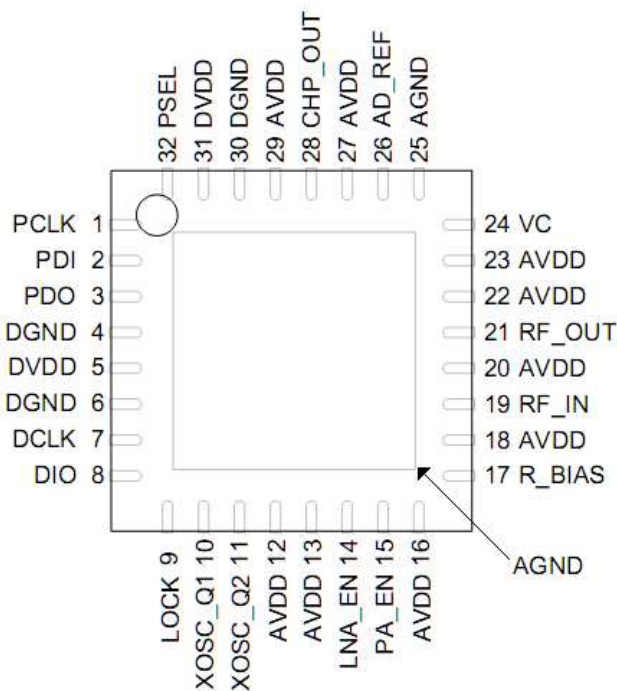


Obr. 4 Doporučené zapojení obvodu

Tab. 9 Přehled součástek doporučeného zapojení obvodu

Součástka	Popis
C1	Impedanční přizpůsobení a stejnosměrné oddělení
C3	Impedanční přizpůsobení a stejnosměrné oddělení
C4	Zatěžovací kondenzátor krystalového oscilátoru
C5	Zatěžovací kondenzátor krystalového oscilátoru
C6	Filtr smyčky fázového závěsu
C7	Filtr smyčky fázového závěsu
C8	Filtr smyčky fázového závěsu
C60	Blokovací kondenzátor
L1	Impedanční přizpůsobení a stejnosměrný pracovní bod CC1020
L2	Impedanční přizpůsobení a stejnosměrný pracovní bod CC1020
R1	Rezistor nastavení proudového zdroje
R2	Filtr smyčky fázového závěsu
R3	Filtr smyčky fázového závěsu
R10	Impedanční přizpůsobení

Obvod se vyrábí pouze v pouzdru QFN 32, které se vyznačuje malými rozměry. Pouzdro nemá klasické nožky, ale pouze pájecí plošky ze své spodní strany. Díky tomu není příliš vhodné pro ruční osazování. Pouzdro včetně rozmístění pinů ukazuje obr. 5. Pozor, pouzdro má ze své spodní strany zemnicí plochu. Pro správnou funkci obvodu je nutné tuto plochu propojit se zemí. Druhotný význam této plochy je pevnější uchycení pouzdra k plošnému spoji. Popis pinů ukazuje tab.10.



Obr. 5 Pouzdro obvodu CC1020, pohled shora

Tab. 10 Popis pinů obvodu CC1020

Pin	Název	Typ <sup>1)</sup>	Popis
-	AGND	AG	Zem analogové části, musí být propojen se zemí
1	PCLK	DI	Hodinový signál SPI
2	PDI	DI	Datový vstup SPI (MOSI)
3	PDO	DO	Datový výstup SPI (MISO)
4	DGND	DG	Zem digitální části
5	DVDD	DP	Napájení digitální části
6	DGND	DG	Zem digitální části
7	DCLK	DO	Hodinový signál datového přenosu pro režim vysílání i příjmu
8	DIO	DIO	Vstup dat při vysílání, výstup dat při příjmu
9	LOCK	DO	Programovatelný výstup
10	XOSC_Q1	AI	Vstup krystalového oscilátoru nebo vnějšího zdroje hodinového signálu
11	XOSC_Q2	AO	Výstup krystalového oscilátoru
12	AVDD	AP	Napájení analogové části
13	AVDD	AP	Napájení analogové části
14	LNA_EN	DO	Programovatelný výstup, může být využit pro řízení napájení vnějšího LNA
15	PA_EN	DO	Programovatelný výstup, může být využit pro řízení napájení vnějšího PA
16	AVDD	AP	Napájení analogové části
17	R_BIAS	AO	Připojení pro vnější rezistor zdroje proudu, 82 k $\Omega$ , tol. 1%
18	AVDD	AP	Napájení analogové části
19	RF_IN	AI	Vstup vysokofrekvenčního přijímaného signálu
20	AVDD	AP	Napájení analogové části
21	RF_OUT	AO	Výstup vysokofrekvenčního vysílaného signálu
22	AVDD	AP	Napájení analogové části
23	AVDD	AP	Napájení analogové části
24	VC	AI	Řídící napětí oscilátoru smyčky fárového závěsu
25	AGND	AG	Zem analogové části
26	AD_REF	AP	Vstup reference 3V pro analogově digitální převodník
27	AVDD	AP	Napájení analogové části
28	CHP_OUT	AO	Výstup kmitočtového detektoru fázového závěsu
29	AVDD	AP	Napájení analogové části
30	DGND	DG	Zem digitální části
31	DVDD	DP	Napájení digitální části
32	PSEL	DI	Aktivační vstup rozhraní SPI, aktivní při log. 0

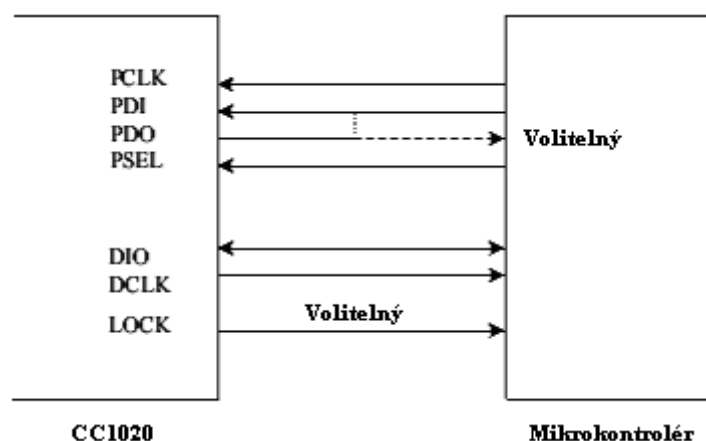
1) Typ pinu je uveden ve zkratkách, kde písmena znamenají:

- A - analogový, analogová část obvodu
- D - digitální, digitální část obvodu
- G - zem
- P - napájení
- I - vstup
- O - výstup

Příklad: DIO = digitální vstup a výstup, AP = napájecí pin analogové části

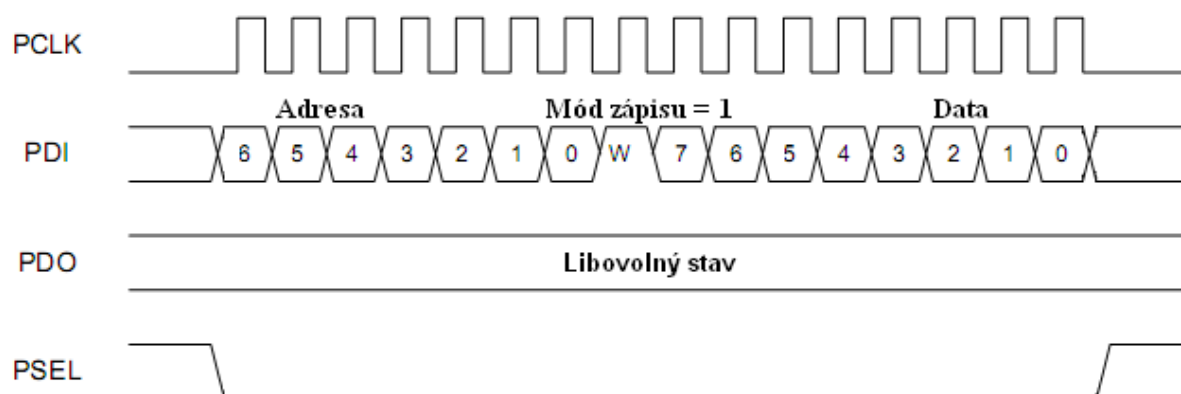
### 3.2 Řídící a datové rozhraní obvodu

Parametry RF transceiveru jsou ovládány pomocí řídicích registrů, které jsou pro řídicí mikrokontrolér přístupné přes sériové rozhraní. Celková struktura rozhraní je na obr. 6. PCLK, PDI, PDO, PSEL jsou signály řídicího rozhraní SPI. Je výhodné, aby mikrokontrolér byl tímto rozhraním vybaven. Zde se obvod chová jako slave. Časování řídicího rozhraní ukazují obr.7 a obr.8. Maximální hodinový kmitočet je 10 MHz.



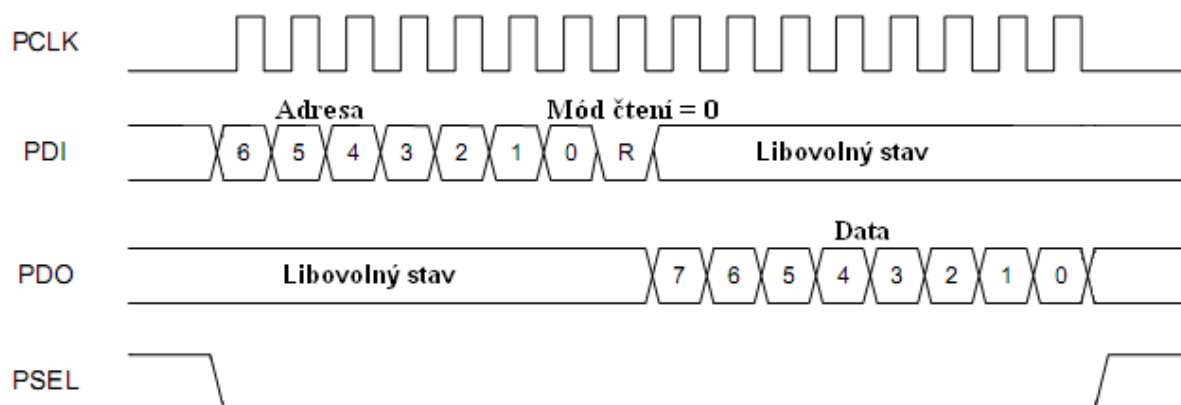
Obr. 6 Rozhraní pro mikrokontrolér

DIO, DCLK, LOCK jsou signály datového rozhraní, kde se obvod chová jako master. Pin DCLK je vhodné propojit s vnějším přerušením mikrokontroléru. Jedná se o hodinový signál datového přenosu, kdy je nutná synchronizace obvodu CC1020 a mikrokontroléru. Přerušení je nastaveno na vzestupnou hranu. Při vysílání při této hraně obvod data čte a mikrokontrolér aktualizuje na následující data. Při příjmu obvod data zapisuje při sestupné hraně, avšak spolehlivé čtení je zaručeno při hraně vzestupné. Časování rozhraní ukazují obr. 9 a obr. 10. Pin DIO slouží k obousměrnému synchronnímu přenosu dat a pinu LOCK lze přiřadit jednu ze šestnácti doplňkových funkcí., viz popis v [3].



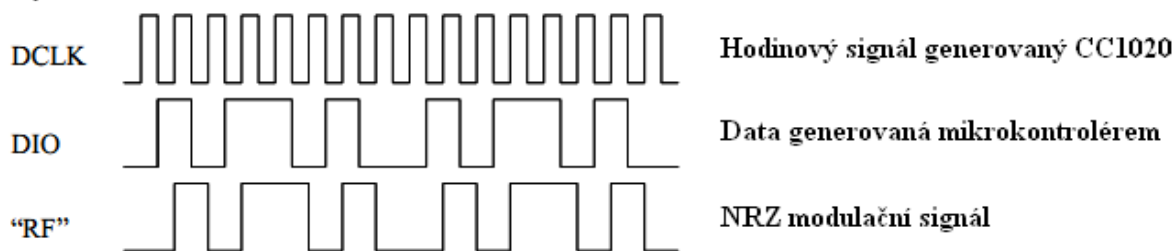
Obr. 7 Časování řídicího rozhraní, zápis do registru



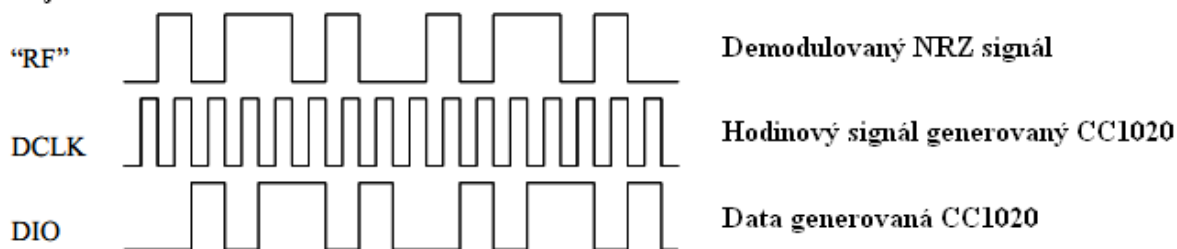


Obr. 8 Časování řídicího rozhraní, čtení z registru

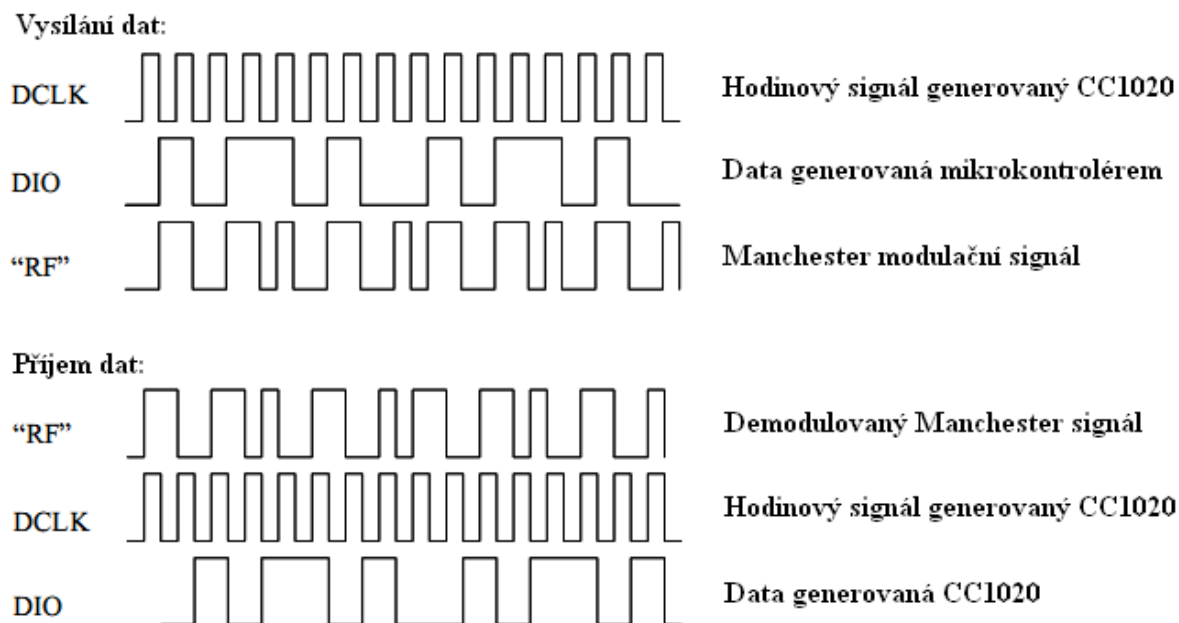
Vysílání dat:



Přijem dat:



Obr. 9 Časování datového rozhraní v NRZ módu



Obr. 10 Časování datového rozhraní v Manchester módu

### 3.3 Řídící registry

Obvod obsahuje 40 řídicích a 11 kontrolních registrů, jejich přehled ukazuje tab. 11. Stojí za povšimnutí, že obvod obsahuje dvě sady registrů pro nastavení pracovního kmitočtu obvodu. Tím je výrazně usnadněna obsluha, protože pro příjem a vysílání na stejném kmitočtu je nutné jiné naladění místního oscilátoru. Při přechodu z příjmu na vysílání se tak pouze přepne registrová sada. Je také možné kmitočtové skákání kdy při aktivní jedné sadě se v sadě druhé připravuje budoucí nastavení kmitočtu. Popis nastavení důležitých registrů bude uveden v dalších kapitolách podle vztahu k tématu.

Registr *MAIN* slouží k přepínání obvodu mezi režimy příjmu a vysílání, k volbě sady kmitočtových registrů a řízení napájení přijímacích a vysílacích obvodů. Přiřazení kmitočtových registrů pro účel příjmu nebo vysílání je libovolný.

Registr *INTERFACE* má význam z hlediska ovládání stavů pinů *LNA\_EN* a *PA\_EN*, kterými se budí spínací tranzistory v rozvodu napájení vnějších zesilovačů modulu.

Tab. 11 Přehled registrů obvodu CC1020

Adresa	Název	Popis	IS <sup>1)</sup>
00h	MAIN	Hlavní řídicí registr	1Fh
01h	INTERFACE	Vlastnosti rozhraní, řízení LNA_EN, PA_EN	01h
02h	RESET	Reset obvodu	FFh
03h	SEQUENCING	Parametry startovací sekvence po resetu	8Fh
04h	FREQ_2A	Nastavení kmitočtu	50h
05h	FREQ_1A	Nastavení kmitočtu	F9h
06h	FREQ_0A	Nastavení kmitočtu	80h
07h	CLOCK_A	Nastavení rychlosti přenosu dat, referenčního kmitočtu PLL	5Ch
08h	FREQ_2B	Nastavení kmitočtu	51h
09h	FREQ_1B	Nastavení kmitočtu	09h
0Ah	FREQ_0B	Nastavení kmitočtu	80h
0Bh	CLOCK_B	Nastavení rychlosti přenosu dat, referenčního kmitočtu PLL	5Ch
0Ch	VCO	Nastavení proudu VCO	44h
0Dh	MODEM	Nastavení módu přenosu, kmitočtu vzorkování demodulátoru	50h
0Eh	DEVIATION	Nastavení zdvihu vysílače	ABh
0Fh	AFC CONTROL	Nastavení AFC, zdvihu přijímače	C6h
10h	FILTER	Nastavení šířky pásma mezifrekvenčního filtru	2Bh
11h	VGA1	Nastavení VGA1	61h
12h	VGA2	Nastavení VGA2	55h
13h	VGA3	Nastavení VGA3	2Eh
14h	VGA4	Nastavení VGA4	29h
15h	LOCK	Nastavení funkce pinu LOCK	20h
16h	FRONTEND	Nastavení napájení obvodů přijímače	78h
17h	ANALOG	Nastavení pásma, parametrů přijímače a vysílače	47h
18h	BUFF_SWING	Nastavení předděliče bufferu	14h
19h	BUFF_CURRENT	Nastavení proudu bufferu	22h
1Ah	PLL_BW	Nastavení šířky pásma filtru PLL	AEh
1Bh	CALIBRATE	Řízení kmitočtové kalibrace obvodu	34h
1Ch	PA_POWER	Nastavení výstupního výkonu PA	03h
1Dh	MATCH	Nastavení kapacitního přizpůsobení vstupu LNA a výstupu PA	00h
1Eh	PHASE_COMP	Kompenzace chyby fáze I a Q	00h
1Fh	GAIN_COMP	Kompenzace chyby zesílení I a Q	00h
20h	POWERDOWN	Řízení spotřeby	00h
21h	TEST1	Test kalibrace PLL	4Dh
22h	TEST2	Test kalibrace PLL	10h
23h	TEST3	Test kalibrace PLL	06h
24h	TEST4	Test nábojové pumpy	00h
25h	TEST5	Test analogově digitálního převodníku	40h
26h	TEST6	Test VGA	00h
27h	TEST7	Test VGA	00h
40h	STATUS	Status obvodu (PLL, kalibrace...)	---
41h	RESET_DONE	Status resetu obvodu	---
42h	RSSI	Indikátor síly přijímaného signálu	---
43h	AFC	Indikátor kmitočtové odchylky přijímaného signálu	---
44h	GAUSS_FILTER	Indikátor offsetu mezifrekvenčního kmitočtu	---
45h	STATUS1	Status PLL	---
46h	STATUS2	Status PLL	---
47h	STATUS3	Status PLL	---
48h	STATUS4	Status analogového převodníku	---
49h	STATUS5	Status filtru kanálu I	---
4Ah	STATUS6	Status filtru kanálu Q	---
4Bh	STATUS7	Status AGC	---

1) Inicializační setup obvodu. Hodnoty registrů jsou generovány pomocí pomůcky *RF Smart Studio*, která je volně ke stažení na stránkách výrobce obvodu. Hodnoty některých registrů jsou upraveny pro potřeby modulu, např. po resetu jsou vypnuté vnější zesilovače PA a LNA. Příklad hodnot je pro pracovní kmitočet 402 MHz, rychlost přenosu 9600 Bd, zdvih  $\pm 4950$  Hz, modulace dat NRZ, modulace nosné GFSK.

### 3.4 Módy přenosu dat

Mód přenosu dat se volí registrem *MODEM*, obvod nabízí tři režimy modulace nosného signálu:

- **asynchronní UART mód** – bitová rychlost vysílaných dat je definována bitovou rychlostí dat přicházejících na pin DIO datového rozhraní obvodu; při příjmu je demodulovaný vzorkovaný signál k dispozici na pinu DIO bez jakékoliv synchronizace; výrobce nedoporučuje používat
- **synchronní Manchester mód** - data jsou obvodem převáděna do Manchester kódu, výsledná bitová rychlost je poloviční proti symbolové.
- **synchronní NRZ mód** - data jsou vysílána v přímém NRZ kódu, výsledná bitová rychlost je rovna rychlosti symbolové.

U synchronního Manchester a NRZ kódu je bitová rychlost přenosu definována nastavením registrů RF transceiveru. Přehled vlastností jednotlivých módů je v tab. 12.

Tab. 12 Módy přenosu dat, vlastnosti

	UART	synchronní Manchester	synchronní NRZ
<b>Rychlost datového toku</b>	definována mikrokontrolérem	generována hardwarově	
<b>Kódování dat</b>	definováno kódem vstupních dat	Manchester kód generován hardwarově	přímá data
<b>Obnova hodin</b>	definována mikrokontrolérem	hardwarově	hardwarově

### 3.5 Rychlost přenosu dat

Rychlost přenosu dat se nastavuje registry *CLOCK\_A* a *CLOCK\_B*, každý přísluší jedné sadě registrů nastavení kmitočtu. Rychlost je závislá na hodnotě referenčního kmitočtu PLL a na přídavných dělitelech tohoto kmitočtu. Referenční kmitočet se vždy volí nejvyšší možný. Vztahy pro výpočet referenčního kmitočtu a rychlosti přenosu jsou uvedeny v [3] a mají tvar:

$$f_{ref} = \frac{f_{xosc}}{REV\_DIV + 1} \quad (3),$$

$$B.R. = \frac{f_{ref}}{8 \cdot DIV1 \cdot DIV2} \quad (4),$$

kde B.R. je datová rychlost, parametr *DIV1* se překóduje na *MCLK\_DIV1* podle tab.13, *DIV2* se překóduje na *MCLK\_DIV2* podle tab.14 a *REV\_DIV*, *MCLK\_DIV1* a *MCLK\_DIV2* odpovídají a jednotlivým bitům registrů *CLOCK\_X* podle tab.15. Vztahy platí pro pásmo 402 – 470 Mhz, jednotky Hz (stejně ve všech dalších vztazích není-li uvedeno jinak)

Tab. 13 Překódovací tabulka DIV1

MCLK_DIV1	DIV1
000	2,5
001	3
010	4
011	7,5
100	12,5
101	40
110	48
111	64

Tab. 14 Překódovací tabulka DIV2

MCLK_DIV2	DIV2
00	1
01	2
10	4
11	8

Tab. 15 Registr CLOCK\_X

Název	REV_DIV	MCLK_DIV1	MCLK_DIV2
Rozsah bitů	7:5	4:2	1:0

Rozsah referenčního kmitočtu má být:

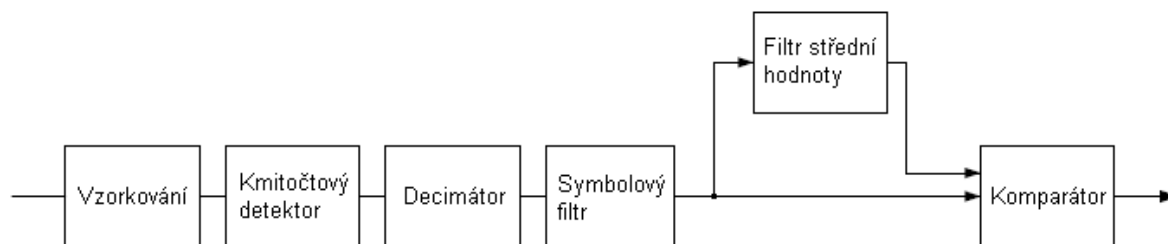
$$9,8304 \geq f_{ref} > \frac{f_c}{256} [MHz] \quad (5)$$

### 3.6 Příjem dat

Přijímač je typu superheterodyn s jedním směřováním. Mezifrekvenční signál je vzorkován a kmitočtový detektor vyhodnocuje jeho okamžitý kmitočet. Výsledek je v decimátoru převzorkován na nižší vzorkovací kmitočet a následně je filtrován. V komparátoru je signál porovnáván s výstupem filtru střední hodnoty signálu. Tím je zvýšena odolnost přijímače proti šumu. Blokové schéma demodulátoru je na obr. 11.

Před příjmem dat musí být filtr střední hodnoty nastaven na střední hodnotu přijímaného signálu. To se provádí příjmem hlavičky, jejíž střední hodnota leží uprostřed mezi „LO“ a „HI“ úrovněmi signálu. Pro NRZ kód to je posloupnost 010101... nebo jiný signál bez

stejnoseměrné složky, pro Manchester kód to jsou libovolná data. Potřebná délka posloupnosti je závislá na podmínkách příjmu. Z popisu protokolu AX25 v kap. 2.1 je zřejmá kompatibilita obvodu z hlediska synchronizace na začátku příjmu.



Obr. 11 Blokové schéma demodulátoru

Přijímač dále disponuje funkcemi automatického doladění kmitočtu AFC (Automatic Frequency Control), automatického vyrovnání citlivosti AGC (Automatic Gain Control), měření intenzity přijímaného signálu RSSI (Received Signal Strength Indicator).

Kmitočet PLL se v režimu příjmu ladí o mezifrekvenční kmitočet níže nebo výše. Při naladění výše budou demodulovaná data invertována. Vztah pro výpočet kmitočtu nosné je uveden v [3], jeho úpravou dostaneme vztah pro výpočet hodnoty kmitočtového registru *FREQ*:

$$FREQ = 32768 \left( \frac{f_c \pm f_{mf}}{f_{ref}} - \frac{3}{4} \right) \quad (6),$$

kde  $f_c$  je kmitočet přijímané nosné. Takto vypočítaná hodnota se posune o jeden bit vlevo a uloží do registru *FREQ*, kde *FREQ\_2X* je nejvýznamnější bajt. Mezifrekvenční kmitočet přijímače je závislý na frekvenci použitého krystalu a volí se co nejblíže kmitočtu 307,2 kHz. Vztah pro jeho nastavení má tvar:

$$f_{mf} = \frac{f_{xosc}}{8 \cdot (ADC\_DIV + 1)} \quad (7),$$

kde *ADC\_DIV[2:0]* jsou bity registru *MODEM*, rozsah 6:4, kódování hodnot dělících poměrů ukazuje tab.16.

Tab. 16 Překódovací tabulka *ADC\_DIV*

ADC_DIV	dělitel
000	nepodporováno
001	4
010	6
011	8
100	10
101	12
110	14
111	16

Přijímači se nastavuje velikost očekávaného kmitočtového zdvihu signálu. Pro modulaci GFSK a kmitočtové zdvihy menší jak 100 kHz se nastavuje polovina vysílaného zdvihu, v ostatních případech je nastavený zdvih přijímače totožný s vysílačem. Vztah pro nastavení má tvar:

$$RXDEV\_M = 3 \cdot \frac{zdvih \cdot 2^{(3-RXDEV\_X)}}{B.R.} \quad (8),$$

kde  $RXDEV\_M[3:0]$  a  $RXDEV\_X[1:0]$  odpovídají bitům registru  $AFC\_CONTROL$ . Platí, že pokud  $RXDEV\_M < 8$ , pak snížíme  $RXDEV\_X$  o jedničku; pokud  $RXDEV\_M > 15$ , pak zvýšíme  $RXDEV\_X$  o jedničku. Rozsah  $RXDEV\_X$  je 0 až 3.

Pro správný příjem je potřeba nastavit šířku pásma mezifrekvenčního filtru. Pro kmitočtovou modulaci platí přibližný vztah:

$$SBW = (B.R. \cdot 2 \cdot zdvih) \cdot K \quad (9),$$

kde  $SBW$  je výsledná šířka pásma modulovaného signálu a  $K$  je koeficient rozšíření šířky pásma vzhledem možnému nepřesnému naladění přijímače. Úpravou vztahu pro šířku pásma, uvedeného v [3], získáme vztah pro určení hodnoty parametru registru:

$$DEC\_DIV = \frac{f_{mf}}{SBW} - 1 \quad (10),$$

kde  $DEC\_DIV[4:0]$  jsou bity registru  $FILTER$ .

### 3.7 Vysílání dat

Syntetizovaný vysokofrekvenční signál postupuje přímo do koncového PA zesilovače. Kmitočtová modulace (FSK) probíhá již při syntéze signálu. Je možné použít Gaussovský filtr datového signálu pro potlačení postranních pásem (modulace GFSK). V případě amplitudové modulace OOK (On-Off Keying) je regulován výkon PA stupně. Kmitočet, zdvih, výkon vysílače a typ modulace jsou definovány hodnotami řídicích registrů.

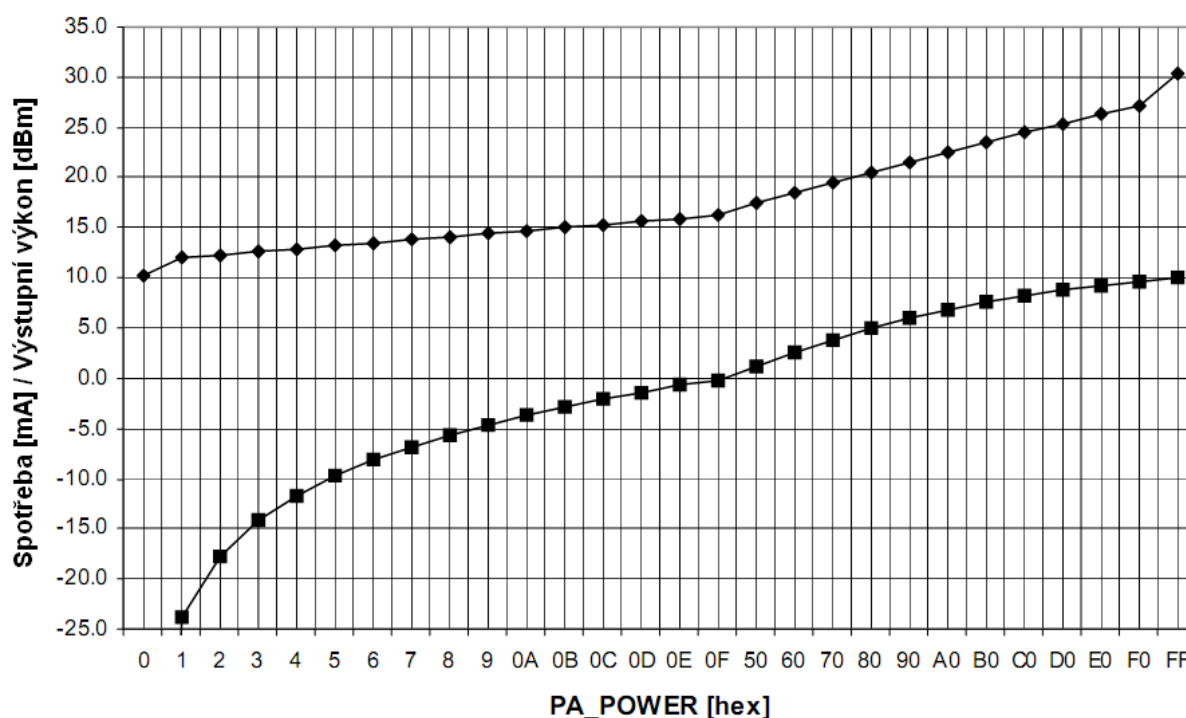
Kmitočet vysílače se ladí na kmitočet nosné signálu. Při vysílání jsou postranní kmitočtové složky přičítány nebo odečítány od nosné podle stavu modulačního signálu. Ve výsledku vysílání nikdy neprobíhá na kmitočtu nosné, ale vždy na jednom z postranních kmitočtů. Vztah pro výpočet nosné vysílače je podobný vztahu pro výpočet ladícího signálu přijímače () a má tvar:

$$FREQ = 32768 \left( \frac{f_c}{f_{ref}} - \frac{3}{4} \right) \quad (11).$$

Takto vypočítaná hodnota se posune, stejně jako v případě přijímače, o jeden bit vlevo a uloží do registru *FREQ*, kde *FREQ\_2X* má nejvyšší význam. Vztah pro výpočet nastavení kmitočtového zdvihu má tvar:

$$TXDEV\_M = \frac{zdvih \cdot 2^{(16-TXDEV\_X)}}{f_{ref}} \quad (12),$$

kde *TXDEV\_M[3:0]* a *TXDEV\_X[2:0]* odpovídají bitům registru *DEVIATION*. Platí, že pokud *TXDEV\_M* < 8, pak snížíme *TXDEV\_X* o jedničku; pokud *TXDEV\_M* > 15, pak zvýšíme *TXDEV\_X* o jedničku. Rozsah *TXDEV\_X* je 0 až 7. Výkon vysílače je nastaven registrem *PA\_POWER*. Registr umožňuje nastavit výkon vysílače ve 32 krocích. Závislost výkonu na hodnotě registru a odpovídající proudovou spotřebu ukazuje obr.12.



Obr. 12 Výkon a spotřeba CC1020 vs. nastavení registru *PA\_POWER*

### 3.8 Kódování dat

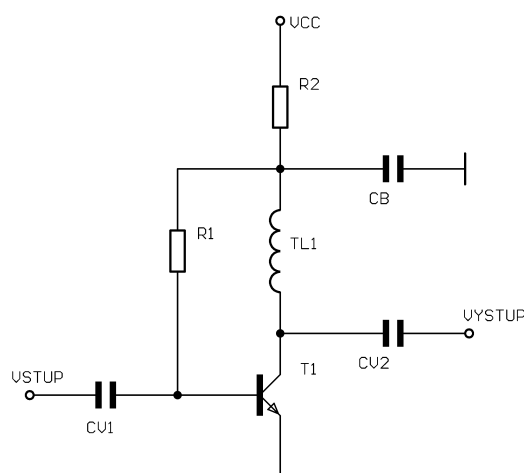
Obvod vysílá data v podobě, v jaké jsou dodávána řídicím mikrokontrolérem. Pouze v režimu Manchester převede datové bity na odpovídající symboly kódu Manchester. Je možné využít funkci scamblování (kódování pseudonáhodnou posloupností) dat s generačním polynomem  $x^9 + x^5 + 1$ . Je-li potřeba data přenášet kódovaná jiným typem kódu, musí tuto funkci zastat řídicí mikrokontrolér.



## 4 Obvodové řešení modulu

### 4.1 UHF zesilovače

Pro maximální jednoduchost bylo voleno zapojení s bipolárními tranzistory pracující ve třídě A se stabilizací pracovního bodu s odporem v kolektoru a napájením kolektorového obvodu přes tlumivku. Toto zapojení má výhodu v absenci emitorového odporu, který zhoršuje šumové číslo tranzistoru a budicí obvod báze má poměrně vysokou impedanci a tedy i méně tlumí vstupní signál. Principiální zapojení je na obr. 13.

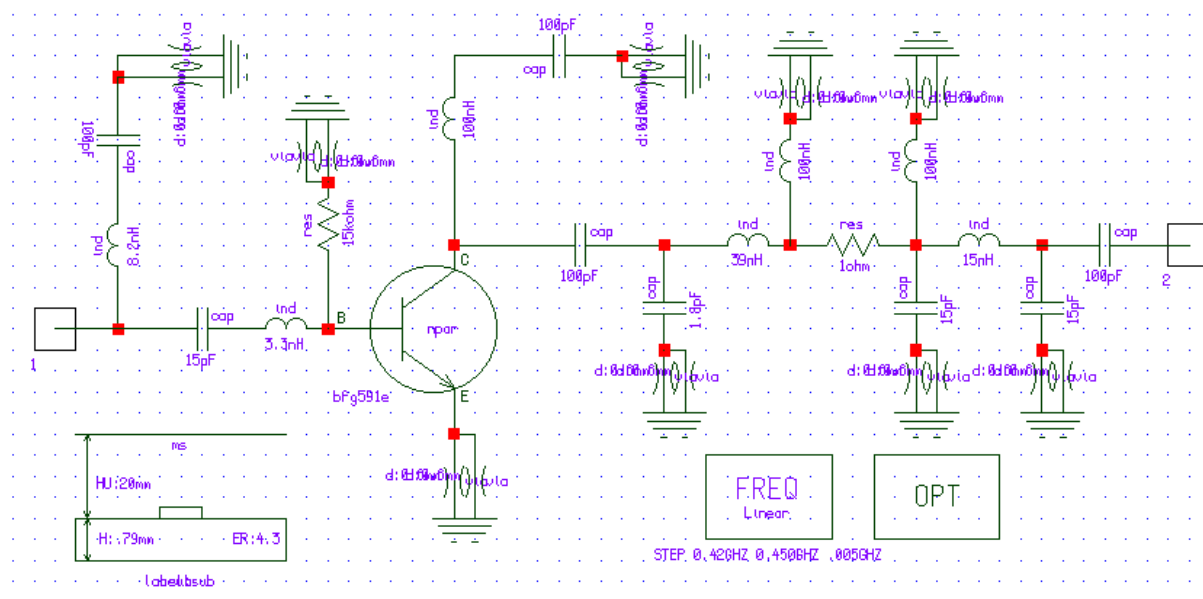


Obr. 13 Principiální schéma zesilovačů

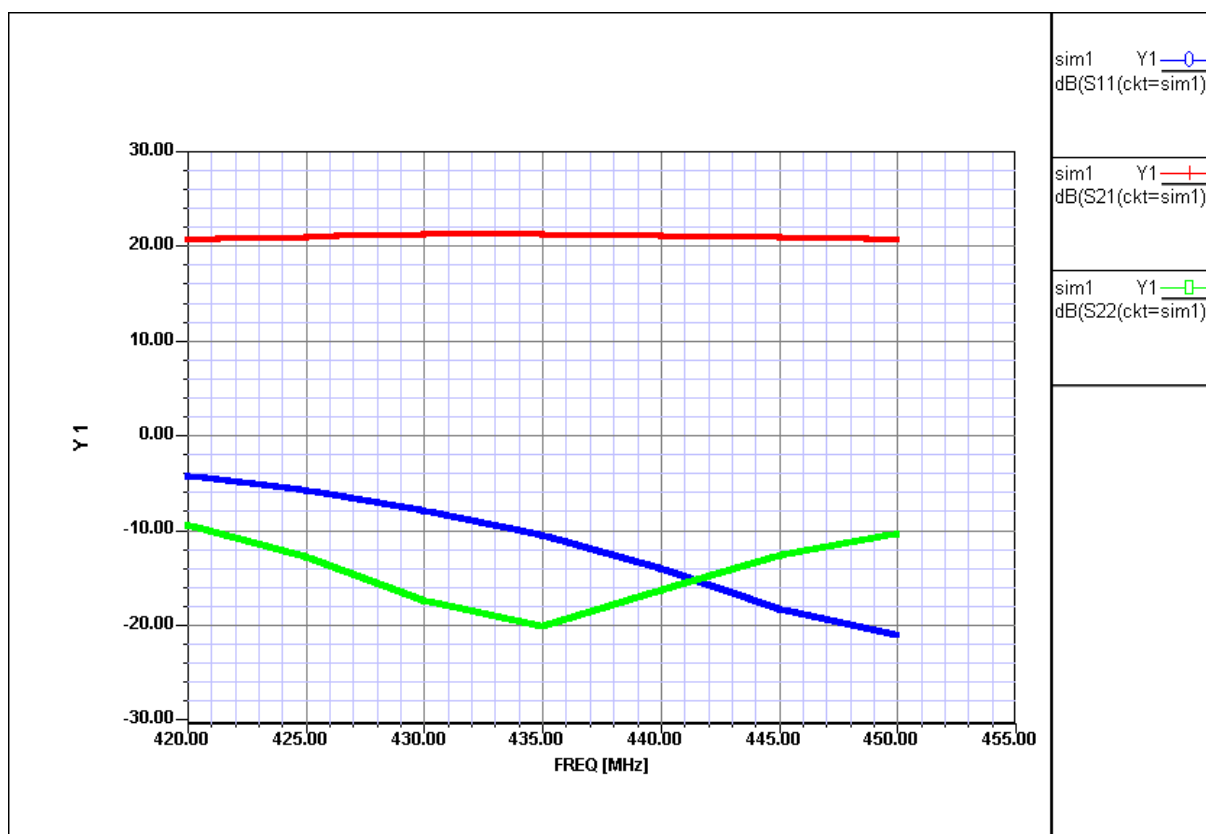
Návrh a simulace zesilovačů byly provedeny v prostředí *Ansoft Serenade 8.5* s ohledem na maximální dosažený zisk a co nejlepší přizpůsobení zesilovače. Do modelu zesilovače byl i zahrnut model přepínače režimu příjem – vysílání a výstupní filtr vyšších harmonických složek.

#### 4.1.1 PA stupeň vysílače

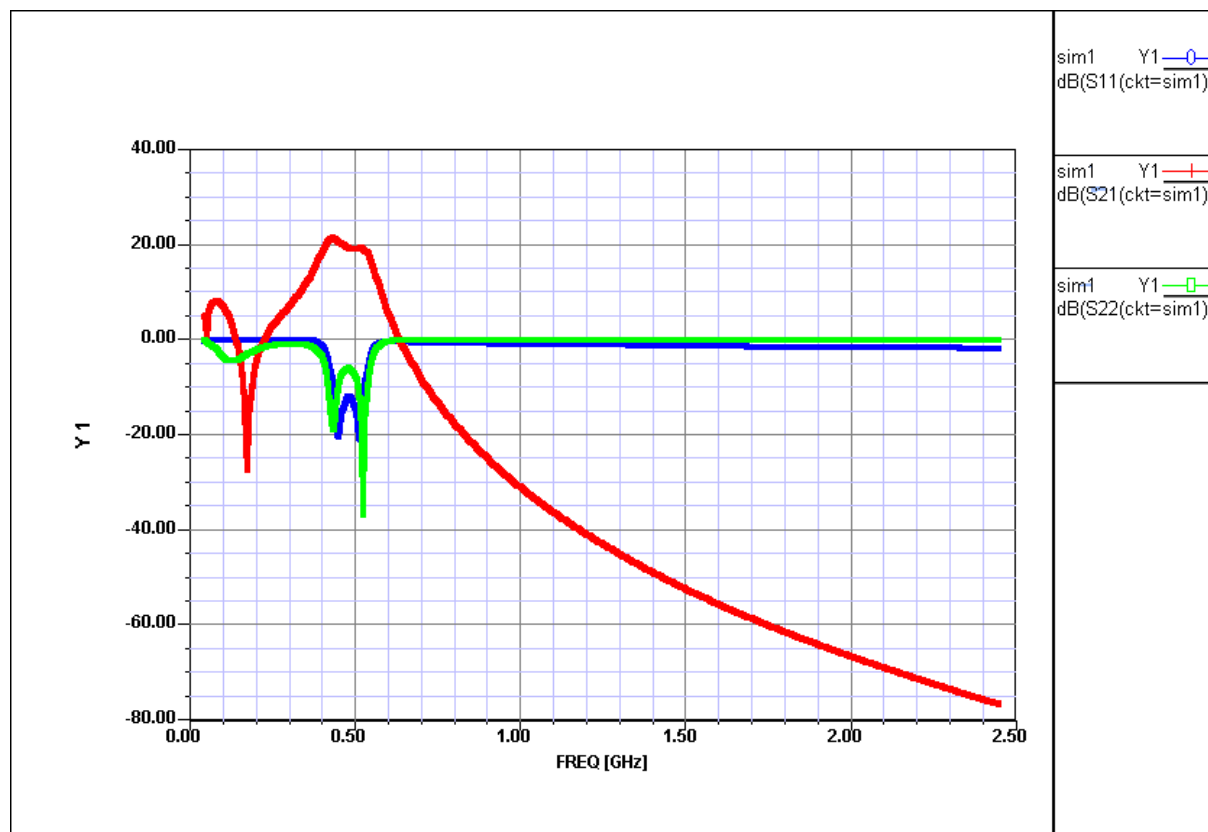
Pro PA stupeň vysílače byl zvolen tranzistor BFG591 pro jeho vhodné výkonové vlastnosti a snadnou dostupnost. Tranzistor má pracovní bod  $U_{CE} = 11\text{ V}$  a  $I_C = 50\text{ mA}$ . Zesilovač je na svém vstupu impedančně přizpůsoben přímo na obvod CC1020. Tím je výsledný přizpůsobovací obvod jednodušší a méně citlivý na tolerance součástek. Výstup zesilovače je přizpůsoben na impedanci  $50\ \Omega$ . Za přizpůsobovacím členem následuje přepínač příjem – vysílání a dále filtr vyšších harmonických složek. Model PA je na obr. 14, přizpůsobení zesilovače pro pracovní kmitočty je na obr. 15 a přizpůsobení pro kmitočty 0 – 2,5 GHz je na obr. 16.



Obr. 14 Model PA



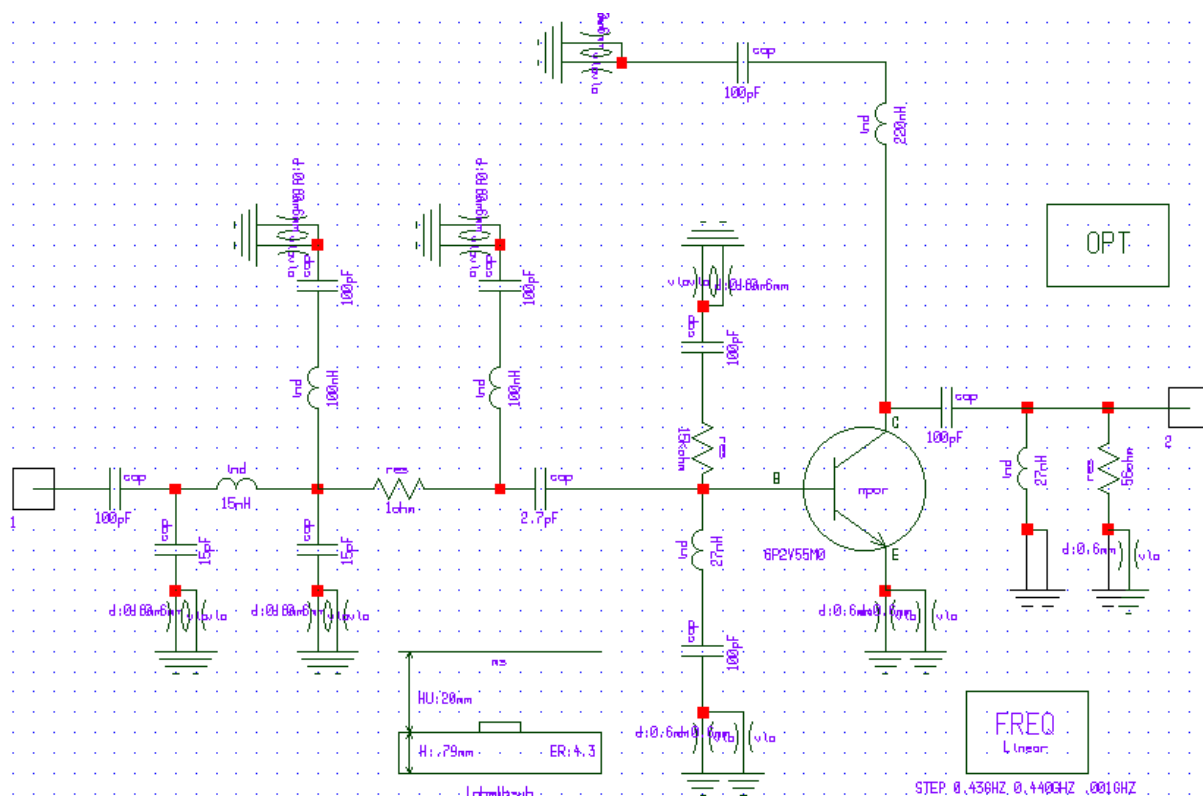
Obr. 15 Přizpůsobení PA pro pracovní kmitočet



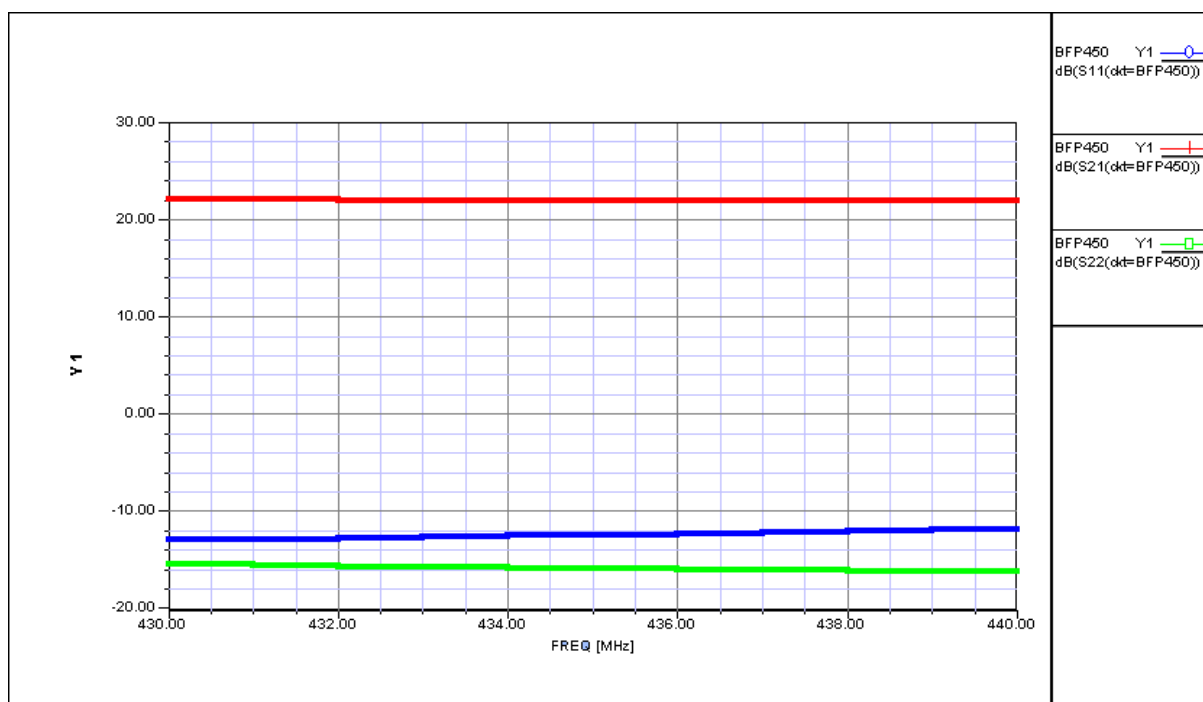
Obr. 16 Přizpůsobení PA pro 0 - 2,5 GHz

#### 4.1.2 LNA stupeň přijímače

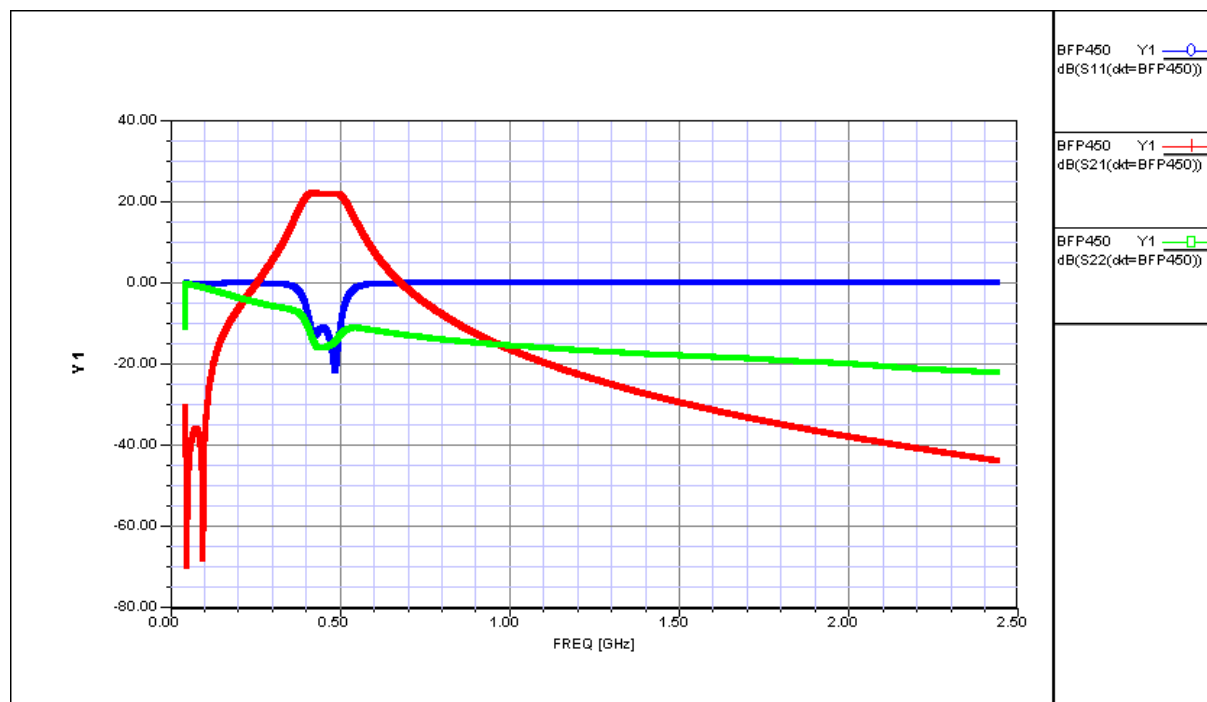
Pro LNA stupeň přijímače byl zvolen tranzistor BFP540 pro jeho vhodné šumové vlastnosti, nenáročnost na šumové přizpůsobení a snadnou dostupnost. Tranzistor má pracovní bod  $U_{CE} = 2,5 \text{ V}$  a  $I_C = 10 \text{ mA}$ . Zesilovač je na svém výstupu impedančně přizpůsoben přímo na obvod CC1020. Tím je výsledný přizpůsobovací obvod jednodušší a méně citlivý na tolerance součástek. Impedance vstupního a výstupního pinu RF části obvodu CC1020 je uvedena výrobcem v [3]. Vstup zesilovače je přizpůsoben na impedanci  $50 \Omega$ . Model LNA je na obr. 17, přizpůsobení zesilovače pro pracovní kmitočty je na obr. 18 a přizpůsobení pro kmitočty 0 – 2,5 GHz je na obr. 19.



Obr. 17 Model LNA



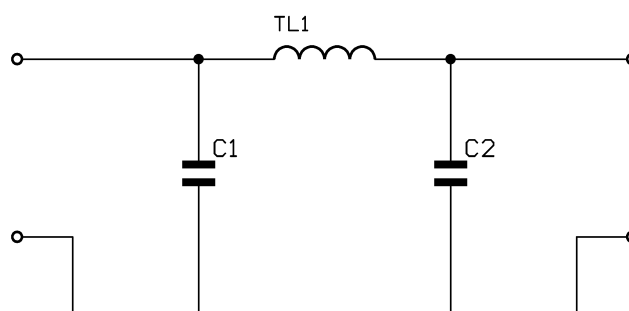
Obr. 18 Přizpůsobení LNA pro pracovní kmitočty



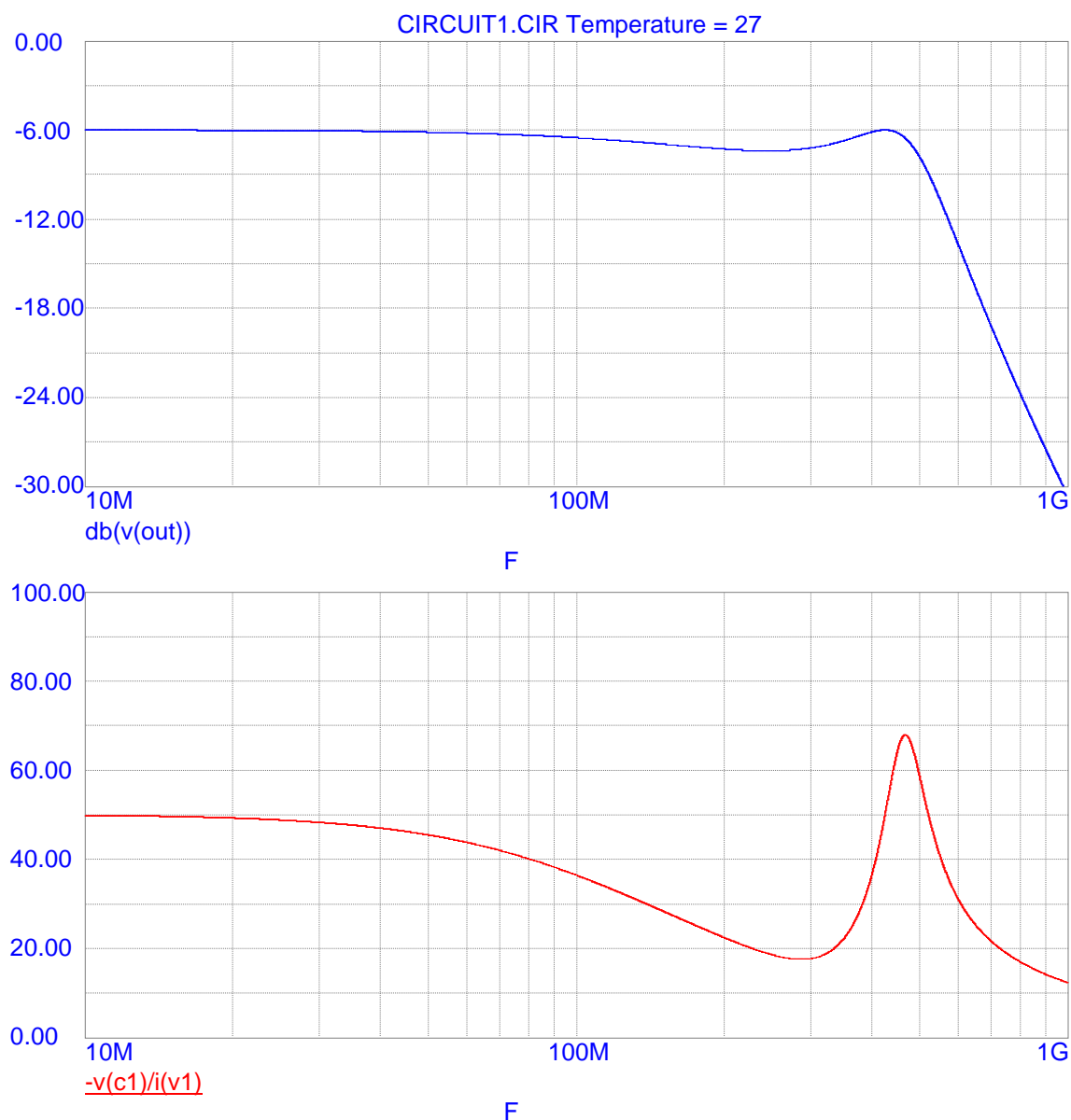
Obr. 19 Přizpůsobení LNA pro 0 - 2,5 GHz

#### 4.1.3 Filtr vyšších harmonických složek

Pro vyšší spektrální čistotu vysílaného a přijímaného signálu je před anténní konektor zařazen filtr vyšších harmonických složek. Návrh filtru byl proveden v prostředí *AADE Filter Design and Analysis*. Byl zvolen filtr typu dolní propust s aproximací Chebyshev se zvlněním v propustném pásmu 1 dB. Schéma filtru je na obr. 20, modulová kmitočtová charakteristika je na horní polovině obr. 21 a průběh vstupní a výstupní impedance je na spodní polovině obr. 21.



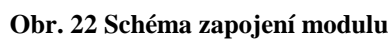
Obr. 20 Filtr vyšších harmonických složek



Obr. 21 Charakteristiky filtru vyšších harmonických

## 4.2 Zapojení modulu

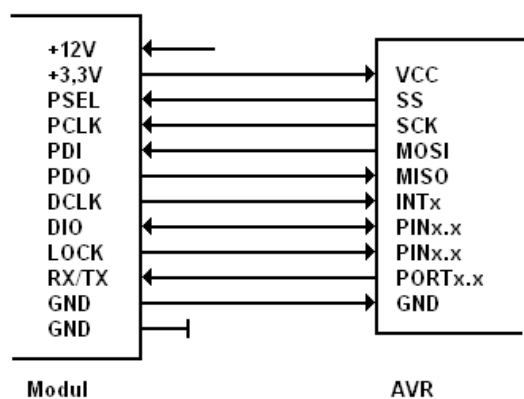
Z důvodu vysoké specializace obvodu CC1020 je použito výrobcem doporučené zapojení. Zapojení se odlišuje pouze vstupním a výstupním přizpůsobovacím obvodem, který byl impedančně upraven pro přizpůsobení k připojeným zesilovačům. Stejnsměrné zapojení přizpůsobovacích obvodů bylo dodrženo dle původního zapojení výrobce. Hodnoty součástek filtru fázového závěsu byly určeny pomocí pomůcky *Smart RF Studio 6.5.1*, která je volně ke stažení na stránkách výrobce obvodu. Podrobné schéma modulu ukazuje obr.22, osazovací výkres se zapojením pinů modulu je v příloze..... Podklady pro výrobu desky jsou součástí přílohy v elektronické podobě.



Deska modulu je osazena součástkami z obou stran. Horní strana obsahuje signálové cesty a součástky a na spodní straně je rozvod napájení a blokovací kondenzátory. Modul je napájen ze zdroje 12V. Modul záměrně neobsahuje řídicí mikrokontrolér, čímž se stává univerzálněji použitelný a dosažitelné vlastnosti modulu tak nejsou ovlivněny pevně vestavěným obvodem. Součástí modulu je zdroj 3,3V pro napájení řídicího mikrokontroléru. Napájecí proud mikrokontroléru by neměl překročit 80 mA

### 4.3 Rozhraní modulu

Obr. 23 ukazuje rozhraní modulu s příkladem připojení řídicího mikrokontroléru Atmel AVR. Rozhraní obsahuje rozhraní obvodu CC1020 (PSEL, PCLK, PDI, PDO, DCLK, DIO, LOCK), které je rozšířeno o ovládání diodového přepínače antény mezi výstupem PA a vstupem LNA (RX/TX) a o rozvod napájení (+12V vstup, +3,3V výstup, GND). Na straně mikrokontroléru jsou uvedeny funkčně odpovídající piny.



Obr. 23 Rozhraní modulu



## 5 Programové řešení

Softwarová část byla rozdělena do dvou modulů. Modul **cc1020.c** a k němu přidružený hlavičkový soubor **cc1020.h** obsahují obecné rutiny pro komunikaci s obvodem CC1020. Bez těchto rutin nelze odvod provozovat. Druhý modul **tnc.c** a **tnc.h** obsahují řešení algoritmů TNC pro packet radio. Moduly jsou napsány v jazyce C pro mikrokontroléry Atmel AVR ATmega128 a ATmega16.

### 5.1 Terminálový program

Jako terminálový program byl zvolen *AGWTerminal* řeckého autora George Rossopoulse. Tento program při své činnosti spolupracuje s *AGW Packet Engine*, oba programy lze stáhnout z [9] a [10]. Tento program samočinně ovládá některé z funkcí TNC, jmenovitě *TxDelay*, *TxTail*, *Persistence* a *Slot\_time*. Způsob ovládání je možně nastavit. *AGW* neumožňuje využívat funkce *Set\_Hardware*, která je v KISS definována jako uživatelská pro. Proto pro ovládání dalších funkcí TNC byla zvolena metoda datové zprávy se speciálním formátem, u kterého je nepravděpodobný výskyt v reálném provozu.

### 5.2 Modul cc1020.c

Struktura a algoritmy modulu byly převzaty z doporučení výrobce obvodu [3] a [4]. Modul se skládá z několika funkcí umožňujících změnu obsahu řídicích registrů a přepínání mezi režimy příjmu a vysílání. Pro správnou funkci samostatného modulu je potřeba definovat funkci pro časová zpoždění.

#### 5.2.1 Konfigurační pole DefaultConfig433

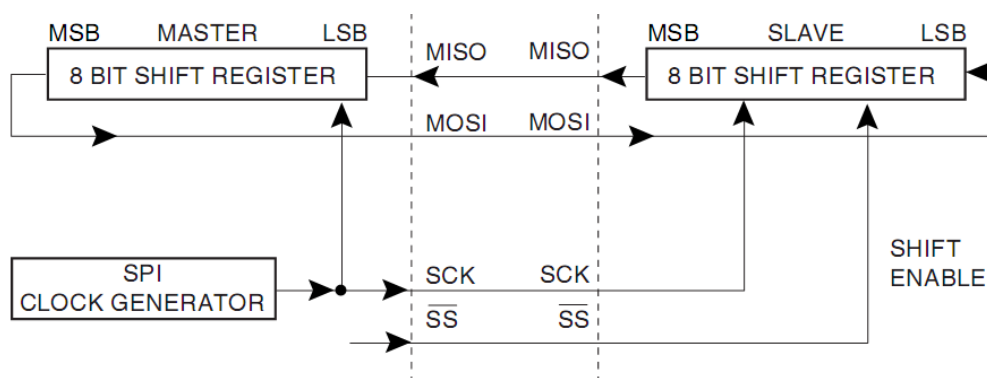
Proměnná typu pole s buňkami rozsahu jednoho bajtu. Pole obsahuje konfiguraci registrů obvodu CC1020, která je po resetu do obvodu přenesena funkcí *setup\_cc1020*. V tomto případě umístění v paměti EEPROM, tím je dosaženo, že si zařízení pamatuje poslední použité nastavení.

### 5.2.2 Funkce `read_cc1020`

Funkce `read_cc1020` slouží ke čtení hodnoty registru obvodu CC1020. Má jeden vstupní parametr, *address*, který vyjadřuje adresu čteného registru. Funkce vrací data v rozsahu jednoho bajtu.

**`unsigned char read_cc1020(unsigned char address);`**

Funkce posune zadanou adresu o jeden bit vlevo, LSB bit adresy nastaví na 0. Tím je pro obvod CC1020 definován režim čtení, viz. obr. 8, kap. 3.2. Protože SPI rozhraní mikrokontroléru AVR aktivuje hodinový signál jen, když se data odesílají, odešle se adresa registr dvakrát. CC1020 po prvním adresovém bajtu připraví data k odeslání, druhý adresový bajt ignoruje a odešle obsah adresovaného registru. Ten se po té vyzvedne z datového registru SPI. Princip SPI ukazuje obr. 24.



Obr. 24 Princip SPI

### 5.2.3 Funkce `write_cc1020`

Funkce `write_cc1020` slouží k zápisu hodnoty do registru obvodu CC1020. Má dva vstupní parametry. *Address* je adresa zapisovaného registru, *data* je zapisovaný obsah.

**`void write_cc1020(unsigned char address, unsigned char data);`**

Funkce posune zadanou adresu o jeden bit vlevo, LSB bit adresy nastaví na 1. Tím je pro obvod CC1020 definován režim zápis, viz. obr. 7, kap. 3.2. Funkce nevrací žádný parametr.

#### 5.2.4 Funkce `setup_cc1020`

Funkce `setup_cc1020` slouží k inicializaci obvodu CC1020 po jeho resetu. Pomocí funkce `write_cc1020` přenese obsah `DefaultConfig433` do registrů CC1020. Nemění se pouze obsah registru `RESET`. Funkce nemá a nevrací žádný parametr.

**`void setup_cc1020_registers(void);`**

#### 5.2.5 Funkce `reset_cc1020`

Funkce `reset_cc1020` slouží k provedení resetu obvodu CC1020 po připojení napájecího napětí. Funkce nemá a nevrací žádný parametr.

**`void reset_cc1020(void);`**

#### 5.2.6 Funkce `calibrate_cc1020`

Funkce `calibrate_cc1020` slouží ke kalibraci kmitočtu smyčky fázového závěsu syntezátoru nosné vlny vysílače a ladícího signálu přijímače. Algoritmus je převzat z [4]. Kalibrace se provádí zvlášť pro vysílací a přijímací stranu. Byla-li kalibrace úspěšná, funkce vrací parametr 1, jinak vrací 0.

**`unsigned char calibrate_cc1020(void);`**

#### 5.2.7 Funkce `setup_cc1020_pd`

Funkce `setup_cc1020_pd` slouží pro úsporu energie při neaktivitě modulu. Obvod CC1020 je při něm uveden do stavu minimální spotřeby a PA a LNA zesilovače jsou vypnuté. Funkce nemá a nevrací žádný parametr.

**`void setup_cc1020_pd(void);`**

#### 5.2.8 Funkce `wakeup_cc1020_rx` a `wakeup_cc1020_tx`

Funkce `wakeup_cc1020_rx` a `wakeup_cc1020_tx` slouží pro zapnutí oscilátoru obvodu CC1020 a čekají na ustálení jeho parametrů. Současně se aktivuje příslušná sada registrů pro

nastavení kmitočtu. Pro přijímač byla zvolena sada A, pro vysílač sada B. Funkce nemají a nevracejí žádný parametr.

```
void wakeup_cc1020_rx(void);  
void wakeup_cc1020_tx(void);
```

#### 5.2.9 Funkce `setup_cc1020_rx`

Funkce `setup_cc1020_rx` slouží jak k zapnutí LNA zesilovače obvodu CC1020, tak k zapnutí LNA zesilovače modulu. Tím se modul uvede do režimu příjmu a na datovém výstupu DIO začne poskytovat demodulovaná data. Pro přepnutí modulu z režimu vysílání je potřeba nejdříve volat funkci `wakeup_cc1020_rx`. Funkce nemá a nevrací žádný parametr.

```
void setup_cc1020_rx(void);
```

#### 5.2.10 Funkce `setup_cc1020_tx`

Funkce `setup_cc1020_tx` slouží jak k zapnutí PA zesilovače obvodu, tak k zapnutí PA zesilovače modulu. Tím se modul uvede do režimu vysílání a podle stavu datového vstupu DIO začne modulovat vysílaný signál. Pro přepnutí modulu z režimu příjmu je potřeba nejdříve volat funkci `wakeup_cc1020_tx`. Funkce má jeden parametr *power*, kterým se nastaví výkon PA. Funkce nevrací žádný parametr.

```
void setup_cc1020_tx(unsigned char power);
```

### 5.3 Modul `tnc.c`

#### 5.3.1 Hlavní funkce `main`

Jádrem programu jsou dva páry kruhových bufferů, ve kterých se tvoří fronty odchozích a příchozích paketů. Každý pár obsahuje buffer pro data (*TX\_Data\_Buffer*) a buffer pro evidenci začátku a konce paketů v datovém bufferu (*TX\_Data\_Packet\_Buffer*). Každý buffer má dvě ukazovátka, jedno slouží pro adresaci při plnění bufferu (ukazovátka *In*), druhé při jeho vyprazdňování (ukazovátka *Out*). Buffery a jejich ukazovátka jsou definovány jako globální proměnné, neboť je k nim vyžadován přístup mnoho funkcemi. Datové buffery mají mít délku alespoň 332 bajtů (jejich délka je libovolně nastavitelná), aby byly schopny pojmout alespoň jeden paket maximální délky. Buffery adres mají délku 8 (pro případ změny musí být délka mocninou čísla 2).

Systém dvou bufferů pro jeden směr přenosu, kdy jeden obsahuje data a druhý počátky a konce paketů, byl zvolen z důvodu, kdy se očekává možnost příchodu více paketů za sebou. Nestačí pouze datový buffer, docházelo by tak se slučování paketů. TNC má za úkol počítat kontrolu cyklického zabezpečení CRC, a to bez znalosti polohy paketu v bufferu není možné. Do adresového bufferu se tak ukládají hodnoty ukazovátka *TX\_Data\_Buffer\_In* po ukončení příjmu paketu. Při odesílání paketu se s buňkou adresového bufferu porovnává stav ukazovátka *TX\_Data\_Buffer\_Out*, shodnost hodnot oznamuje konec paketu v datovém bufferu.

Princip funkce programu je naznačen na obr.25, kde dvojité šipky ukazují tok dat a čárkovaná čára ukazuje vazbu ukazovátka datového bufferu na paměť začátků a konců paketů. Práci s pakety, buffery a ukazovátky ukazuje tab.17.

Po resetu, stav 0 (tab.17) a inicializaci proměnných a modulu jsou všechna ukazovátka vynulována. V adresovém bufferu je uložena adresa počátku nového paketu, na jehož příchod z PC se nyní čeká.

Po příchodu paketu z PC, stav 1, *TX\_Data\_Buffer\_In* ukazuje na třetí prázdnou buňku datového bufferu, nechává se místo o velikosti 2 bajty na kontrolní součet. Tato adresa se uloží do bufferu adres a *TX\_Data\_Packet\_Buffer\_In* se zvýší o jedničku. Nestejná adresa začátku a konce adresového bufferu znamená, že je ve frontě k odeslání jeden nebo více paketů. Počet paketů je roven rozdílu ukazovátek. Nyní se zjišťuje typ přijatého paketu. Není-li to paket datový, provede se změna nastavení parametrů TNC. Je-li paket datový, volá se funkce *Check\_TNC*, která zkoumá, zda se jedná o data k odeslání nebo o konfiguraci TNC. V příkladu se jedná o data k odeslání.

Stav 2 ukazuje situaci v průběhu vysílání paketu. *TX\_Data\_Packet\_Buffer\_Out* se před odesláním zvýšil o jedničku, shodností ukazovátek je indikováno, že nejsou další data k odeslání. *TX\_Data\_Buffer\_Out* ukazuje přibližně do jedné třetiny délky paketu, který se právě vysílá.

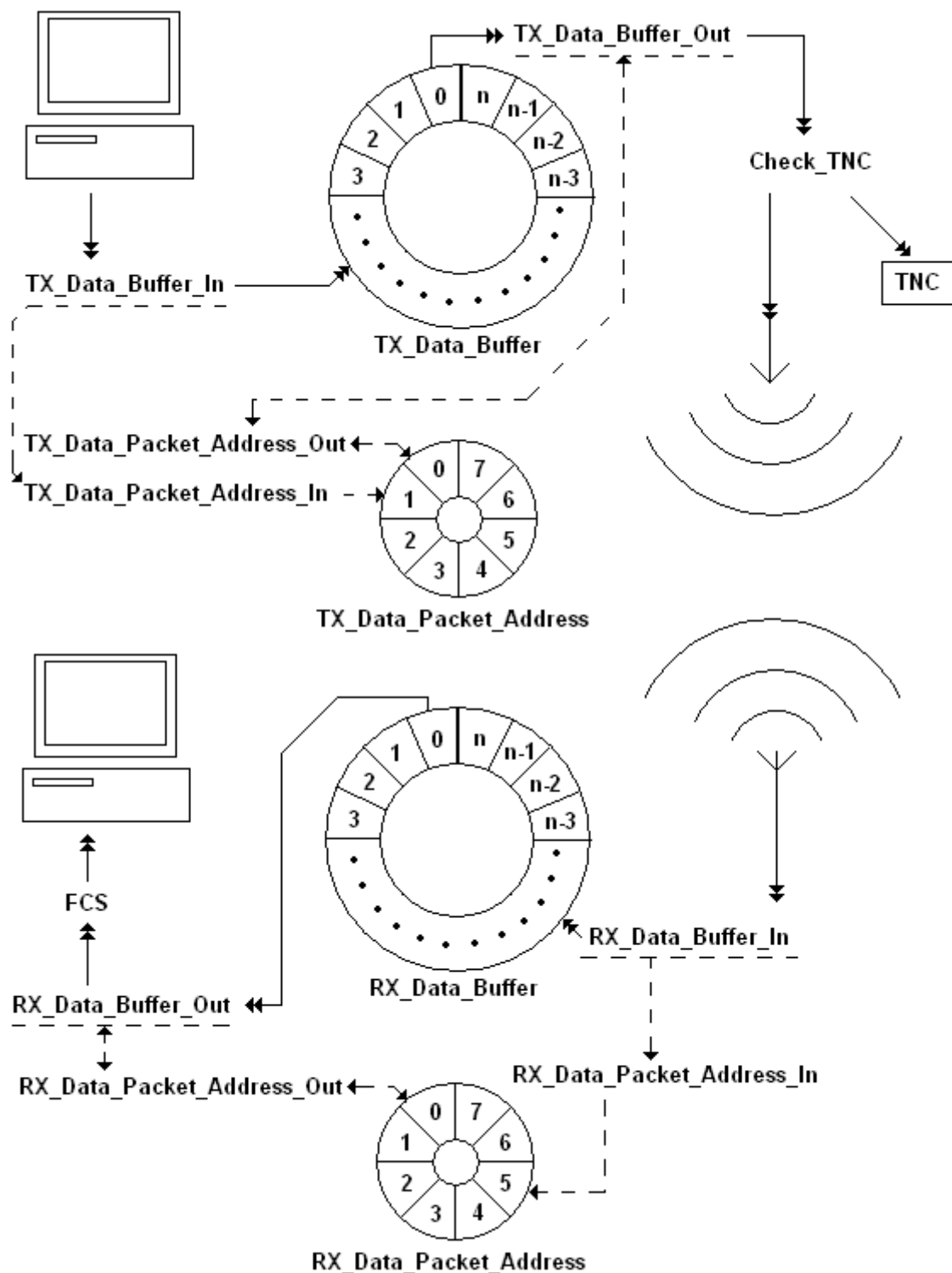
Po odvysílání paketu, stav 3, mají ukazovátka datového bufferu shodnou hodnotu, tj. buffer je prázdný, a ukazují na stejnou adresu jako adresový buffer. TNC je ve situaci podobné stavu 0.

Stav 4 ukazuje situaci, kdy dorazil druhý paket, podobně jako ve stavu 1.

Stav 5 ukazuje situaci, kdy v průběhu vysílání druhého paketu dorazil z PC paket třetí. *TX\_Data\_Buffer\_In* ukazuje na začátek nového paketu, tato adresa je uložena v adresovém bufferu na pozici 3. *TX\_Data\_Buffer\_Out* ukazuje přibližně do dvou třetin právě vysílaného paketu. Na konec právě vysílaného paketu ukazuje *TX\_Data\_Packet\_Buffer\_Out*, tj. pozice 2 adresového bufferu.

Stav 6 ukazuje situaci, kdy byl druhý paket právě odeslán, tj. výstupní ukazovátka datového bufferu dosáhlo adresy konce druhého paketu. Nyní však nenastalo vyprázdnění bufferů, což indikují nestejné údaje ukazovátek obou bufferů. TNC se v podstatě nachází ve stavu 1 nebo stavu 4.

Ve stavu 7 je ukončeno odeslání třetího paketu, TNC se nachází ve stavu podobném stavu 0 nebo stavu 3 a čeká na další pakety z PC.



Obr. 25 Princip funkce TNC

V režimu příjmu je způsob práce s daty obdobný. Dorazí-li paket po rádiové lince, TNC spočítá kontrolu cyklického zabezpečení a podle jejího výsledku rozhodne, zda se data zahodí nebo odešlou do PC, kde se zobrazí terminálovým programem.

Tab. 17 Ukázka práce s buffery, pakety a ukazovátky

Stav	TX_Data_Buffer		TX_Data_Packet_Address				Poznámka
	In <sup>1)</sup>	Out <sup>2)</sup>	In <sup>3)</sup>		Out <sup>4)</sup>		
			U <sup>5)</sup>	B <sup>6)</sup>	U <sup>5)</sup>	B <sup>6)</sup>	
0	0	0	0	0	0	0	Výchozí stav po resetu, buffery jsou prázdné
1	28	0	1	28	0	0	Z PC dorazil první paket k odvysílání
2	28	10	1	28	1	28	Paket je již částečně odeslán, vyílání pokračuje
3	28	28	1	28	1	28	Paket je zcela odeslán, buffery jsou prázdné
4	289	28	2	289	1	28	Z PC dorazil druhý paket k odvysílání
5	311	203	3	311	2	289	Druhý paket je již částečně odeslán, z PC dorazil třetí paket
6	311	289	3	311	2	289	Druhý paket je zcela odeslán, třetí paket je stále ve frontě
7	311	311	3	311	3	311	Třetí paket je zcela odeslán, buffery jsou prázdné

- 1) = ukazovátka TX\_Data\_Address\_In  
 2) = ukazovátka TX\_Data\_Address\_Out  
 3) = ukazovátka TX\_Data\_Packet\_Address\_In  
 4) = ukazovátka TX\_Data\_Packet\_Address\_Out  
 5) stav ukazovátka  
 6) stav buňky bufferu s adresou ukazovátka

### 5.3.2 Funkce Check\_TNC

Funkce Check\_TNC je volána v okamžiku, když dorazí datový paket z PC (KISS). Má za úkol zjistit, zda se jedná o data určená k odeslání rádiovou linkou nebo zda se jedná o konfigurační příkaz pro TNC. Funkce nemá a nevrací žádný parametr.

**void Check\_TNC(void);**

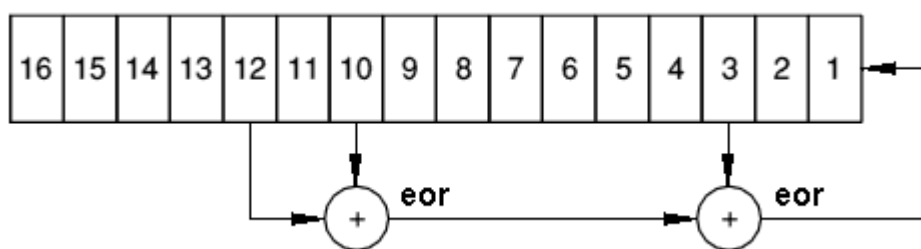
Funkce zkoumá paket a hledá začátek datového pole. Pokud prvních šest bajtů datového pole obsahuje „MY TNC“, jedná se o konfigurační příkaz. V případě, že se jedná o data k odeslání se volá funkce *FCS\_TX*, která vypočítá cyklický redundantní součet paketu. Pokud se momentálně nepřijímá žádný paket, volá se funkce *Get\_Random* a její výsledek se porovnává s parametrem *Persistence*, který je definován jako globální proměnná. Je-li generované číslo menší, volá se funkce *Transmit\_data*, která spustí odvysílání paketu. Pokud je generované číslo větší, čeká se po dobu definovanou parametrem *Slot time*. Po uplynutí této doby se opět generuje náhodné číslo a děj se opakuje, dokud se nepodaří paket odvysílat.

V případě, že se jedná o konfigurační příkaz, funkce přečte následující dva bajty, které kódují typ příkazu a zbývající část paketu převede na číselnou hodnotu v rozsahu 32 bitů. Převod končí v okamžiku, kdy je nalezen znak konce řádku, tj. 0Dh. Toto 32 bitové číslo je pak parametrem příkazu. Funkce hlídá mezní hodnoty parametrů, případě jejich překročení se stávající nastavení nemění.

### 5.3.3 Funkce Get\_Random

Funkce *Get\_Random* slouží ke generování náhodného čísla. Princip generace náhodného čísla ukazuje obr.26. Jedná se o posuvný registr s délkou 16 bitů a zavedenou zpětnou vazbou. Posuvný registr se na každé zavolání funkce posune o jeden bit vlevo. Funkce nemá žádný parametr a vrací vygenerované číslo v rozsahu 8 bitů.

**unsigned char Get\_Random(void);**



Obr. 26 Schéma generátoru náhodného čísla

### 5.3.4 Funkce FCS\_TX

Funkce *FCS\_TX* slouží k výpočtu cyklického redundantního součtu paketu, který je ve frontě k odeslání. Algoritmus výpočtu byl převzat z [12]. Výpočet se provádí z obsahu celého rámce, podle generačního polynomu (1). Vypočítaná hodnota se bitově invertuje a uloží do datového bufferu na konec paketu. Funkce nemá a nevrací žádný parametr.

**void FCS\_TX(void);**

### 5.3.5 Funkce FCS\_RX

Funkce *FCS\_RX* slouží k výpočtu cyklického redundantního součtu paketu, který je ve frontě přijatých paketů. Algoritmus je stejný jako u funkce *FCS\_TX*. Výpočet se provádí z obsahu celého přijatého rámce včetně jeho pole cyklického zabezpečení, podle generačního polynomu (1). Pokud přijatý rámec dorazil bez chyby, je výsledek výpočtu roven 0F47h. Funkce nemá žádný parametr a vrací výsledek výpočtu v rozsahu 16 bitů.

**unsigned int FCS\_RX(void);**



### 5.3.6 Funkce `setup_pd`

Funkce `setup_pd` slouží k vypnutí modulu transceiveru. Obvod modulu se uvede do režimu nízké spotřeby a zakáže se vnější přerušení od hodinového signálu datového rozhraní obvodu. Funkce tak umožňuje snížit zatížení jádra mikrokontroléru v okamžiku, kdy není potřeba komunikace před rádiové rozhraní. Funkce nemá a nevrací žádný parametr.

```
void setup_pd(void);
```

### 5.3.7 Funkce `Transmit_data`

Funkce `Transmit_data` slouží k přepnutí TNC do režimu vysílání. Funkce přepne anténní přepínač do polohy vysílání, aktivuje napájení PA modulu, přepne obvod modulu do režimu vysílání a povolí vnější přerušení. Funkce nemá a nevrací žádný parametr.

```
void Transmit_data(void);
```

### 5.3.8 Funkce `Receive_data`

Funkce `Receive_data` slouží k přepnutí TNC do režimu příjmu. Funkce přepne anténní přepínač do polohy příjem, aktivuje napájení LNA modulu, přepne obvod modulu do režimu příjmu a povolí vnější přerušení. Funkce nemá a nevrací žádný parametr.

```
void Receive_data(void);
```

### 5.3.9 Obsluha vnějšího přerušení `RF_data_clock`

Obsluha vnějšího přerušení `RF_data_clock` slouží odesílání vysílaných dat do modulu v režimu vysílání a k přijímání demodulovaných dat v režimu příjmu. Přerušení je spouštěno vzestupnou hranou hodinového signálu datového rozhraní modulu. Při každém přerušení se zpracuje jeden datový bit. Funkce obsluhy přerušení nemá a nevrací žádný parametr.

```
interrupt [DCLK_INT] void RF_data_clock(void);
```

V režimu vysílání se na začátku paketu po dobu `TxDelay` vysílají hlavičky 7Eh (AX25). Po poslední odeslaném bajtu úvodní hlavičky paketu se začnou vyzvedávat data z datového bufferu. V této části paketu se na data aplikuje bit stuffing, po pěti odeslaných jedničkách v řadě se vloží nula. Na konci paketu se po dobu `TxTail` vysílá koncová hlavička paketu. Po ukončení vysílání se TNC přepne do režimu příjmu.

Každý odesílaný bajt dat (hlavička, data) se převádí na posloupnost bitů. Jako první se vysílá bit s nejnižším významem, pak se provede rotace právě zpracovávaného bajtu dat o jeden bit vpravo. Po osmi rotacích se vyzvednou nová data (hlavička, data) a celý cyklus se opakuje dokud není paket odeslán. K převedení datového bitu na port mikrokontroléru slouží funkce *RF\_data\_encode*, která zajišťuje i kódování NRZI a scamblování.

V režimu příjmu se při každém vyvolaném přerušení z portu mikrokontroléru čte demodulovaný datový bit. Text dále předpokládá situaci, kdy se čeká na nový paket. Pro každý z portu čtený bit se volá funkce *RF\_data\_decode*, která zajišťuje dekódování NRZI a descamblování dat. Následně se testuje, zda posledních sedm přijatých bitů tvoří posloupnost 0111110. Pokud ano, poslední bit se zahodí, zůstane tak posloupnost 0111111. Tato funkce se označuje jako bit stuffing. Pokud bit stuffing nenastal, testuje se posloupnost posledních osmi bitů na hlavičku paketu AX25, hodnota 7Eh. Je-li hlavička nalezena, přijímač se bajtově synchronizuje a od tohoto okamžiku se k přijímaným datům přistupuje po bajtech. Očekává se, že další přijatý bajt bude opět hlavička - protože se na začátku paketu vysílá posloupnost hlaviček. Pokud se tak nestane, byla zaznamenána hlavička důsledkem příjmu šumu pozadí, bajtová synchronizace se zruší a čeká se na novou hlavičku. Pokusy bylo zjištěno, že při příjmu šumu mohou být detekovány dvě a výjimečně i tři hlavičky za sebou, proto byla stanovena podmínka detekce počátku paketu na příjem deseti hlaviček. Adresová část paketu začíná prvním přijatým bajtem o hodnotě jiné jak 7Eh. Druhou možností je přijatý bajt s hodnotou 7Eh se současně detekovaným bit stuffingem. Bajty od adresového pole paketu AX25 až po pole CRC se uloží do datového bufferu přijímače. Koncová hlavička paketu se detekuje příjmem bajtu s hodnotou 7Eh, u které nebyl detekován bit stuffing.

#### 5.3.10 Funkce *RF\_data\_encode*

Funkce *RF\_data\_encode* slouží ke scamblování, podle generačního polynomu (2), vysílaných dat a ke kódování bitové posloupnosti do kódu NRZI. Funkce má vstupní parametr *data*, který má rozsah jeden bajt. Zpracovává se pouze bit s nejnižším významem, toto řešení ušetřilo jednu datovou proměnnou. Výstup funkce je v podobě modulačního signálu na portu mikrokontroléru.

**void *RF\_data\_encode*(unsigned char data);**

#### 5.3.11 Funkce *RF\_data\_decode*

Funkce *RF\_data\_decode* slouží k descamblování přijímaných dat a k dekódování bitové posloupnosti do kódu NRZ. Funkce má vstupní parametr *data*, který má rozsah jeden bajt. Zpracovává se pouze bit s nejvyšším významem. Funkce vrací data v rozsahu jeden bajt, pro další zpracování se využívá pouze nejvýznamnější bit.

**unsigned char *RF\_data\_decode*(unsigned char data);**

### 5.3.12 Funkce *Send\_Kiss\_To\_PC*

Funkce *Send\_Kiss\_To\_PC* slouží k odeslání jednoho paketu z datového bufferu přijímače do PC. Funkce postupně načítá data z bufferu a sestavuje paket KISS protokolu. V případě speciálních znaků provádí jejich překódování, viz kap. 2.2. Při činnosti je volána funkce *Send\_Byte\_To\_UART*. Funkce nemá a nevrací žádný parametr.

**void Send\_Kiss\_To\_PC(void);**

### 5.3.13 Funkce *Send\_Byte\_To\_UART*

Funkce *Send\_Byte\_To\_UART* slouží k odeslání jednoho datového bajtu zkrz rozhraní UART mikrokontroléru. Funkce má parametr *data*, který představuje odesílaná data, funkce nevrací žádný parametr.

**void Send\_Byte\_To\_UART(char data);**

### 5.3.14 Obsluha přerušení příjmu ze sériové linky *Receive\_Kiss\_From\_PC*

Funkce *Receive\_Kiss\_From\_PC* obsluhy přerušení ze sériové linky slouží k příjmu paketů KISS z PC. Funkce dekóduje paket KISS a přijatá data ukládá do datového bufferu vysílače. V případě speciálních znaků provádí jejich překódování, viz kap. 2.2. Funkce nemá a nevrací žádný parametr.

**interrupt [UART\_RXC\_INTERRUPT] void Receive\_Kiss\_From\_PC(void);**

### 5.3.15 Obsluha přerušení časovače *Timer1\_over*(void)

Obsluha přerušení časovače *Timer1\_over*(void) slouží k obsluze systémového času TNC. Funkce při každém přerušení naplní registr časovače tak, aby další přerušení nastalo za 10 ms. Systémový čas se využívá pro odměřování zpoždění dlouhých časů.

**interrupt [TIM1\_OVF] void Timer1\_over(void);**

## 6 TNC CC1020

### 6.1 Parametry TNC

#### Technické údaje:

Rozsah pracovních kmitočtů:	402 – 470 MHz
Rzsah pracovních zdvihů:	$\pm 2$ až $\pm 50$ kHz
Rozsah výstupních výkonů vysílače:	0 – 25 dBm ve 3dB krocích
Zisk PA:	15 dB
Citlivost přijímače:	-90 dBm
Maximální výkon na vstupu přijímače:	-10 dBm
Zisk LNA:	20 dB
Modulační rychlosti:	1200, 2400, 4800, 9600, 19200 Bd
Referenční kmitočet PLL:	4,9152 MHz
Komunikační formát s PC:	19200 Bd, 8 datových bitů, 1 stop bit, bez parity
Rozsah teplot:	-40 až +85 °C
Pracovní poloha:	libovolná
Terminálový program:	AGW packet engine, AGW terminal

#### Provozní omezení:

Protože CC1020 neobsahuje obvody pro potlačení zrcadlových kmitočtů, nesmí být na kmitočtu o 614,4 kHz níže než na naladěném kanále provoz.

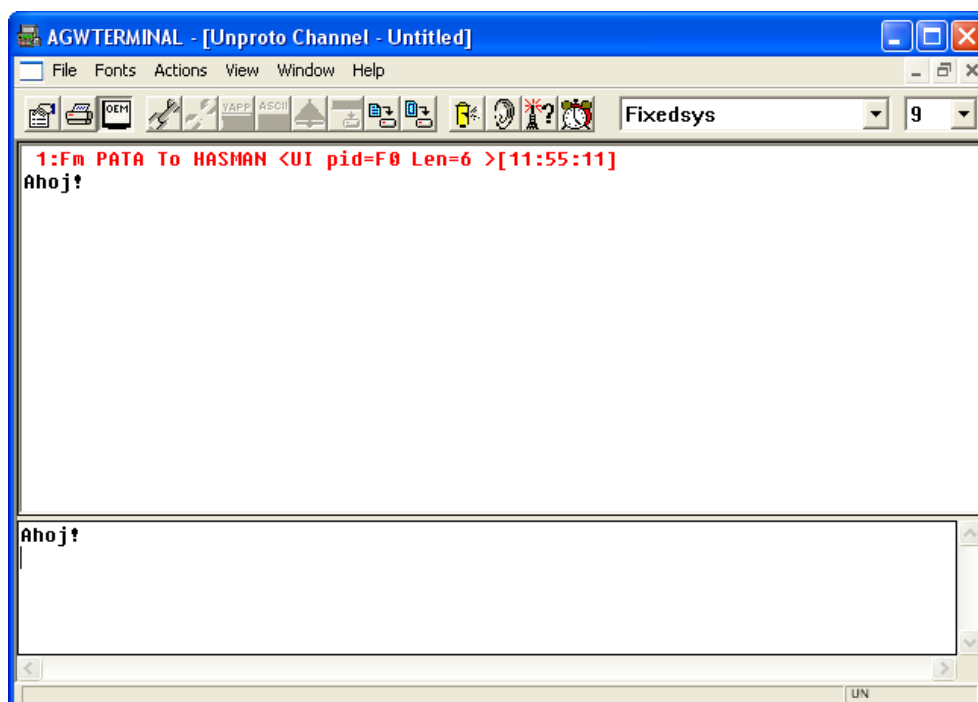
Při provozu na kmitočtech, které jsou vyššími harmonickými složkami referenčního kmitočtu oscilátoru PLL může docházet ke snížení citlivosti přijímače. Jedná se o 82. až 95. harmonickou složku.

Pro každou zvolenou modulační rychlost přijímač vyžaduje určitý minimální zdvih kmitočtové modulace. Tuto závislost uvádí tab. 1. Vyhovuje-li provoz pouze jako vysílač, je možné používat modulační zdvihy nižší.

### 6.2 Návod k obsluze

Komunikační okno terminálového programu ukazuje obr.27. V jeho větší horní polovině se chronologicky zobrazují přijaté a odeslané pakety. Zobrazení je barevně rozlišeno na

pakety komunikace uživatele modrou barvou a na ostatní komunikaci červenou barvou. Do spodní menší části se píše zprávy. Celkově je ovládání programu intuitivní.



Obr. 27 Okno zprávy terminálového programu AGW

TNC podporuje řadu konfiguračních příkazů, které se posílají jako speciálně formátovaná zpráva. Příkazy nejsou závislé na nastavených volacích značkách odesílatele nebo příjemce. Seznam příkazů je v tab.18, příkazy mají formát:

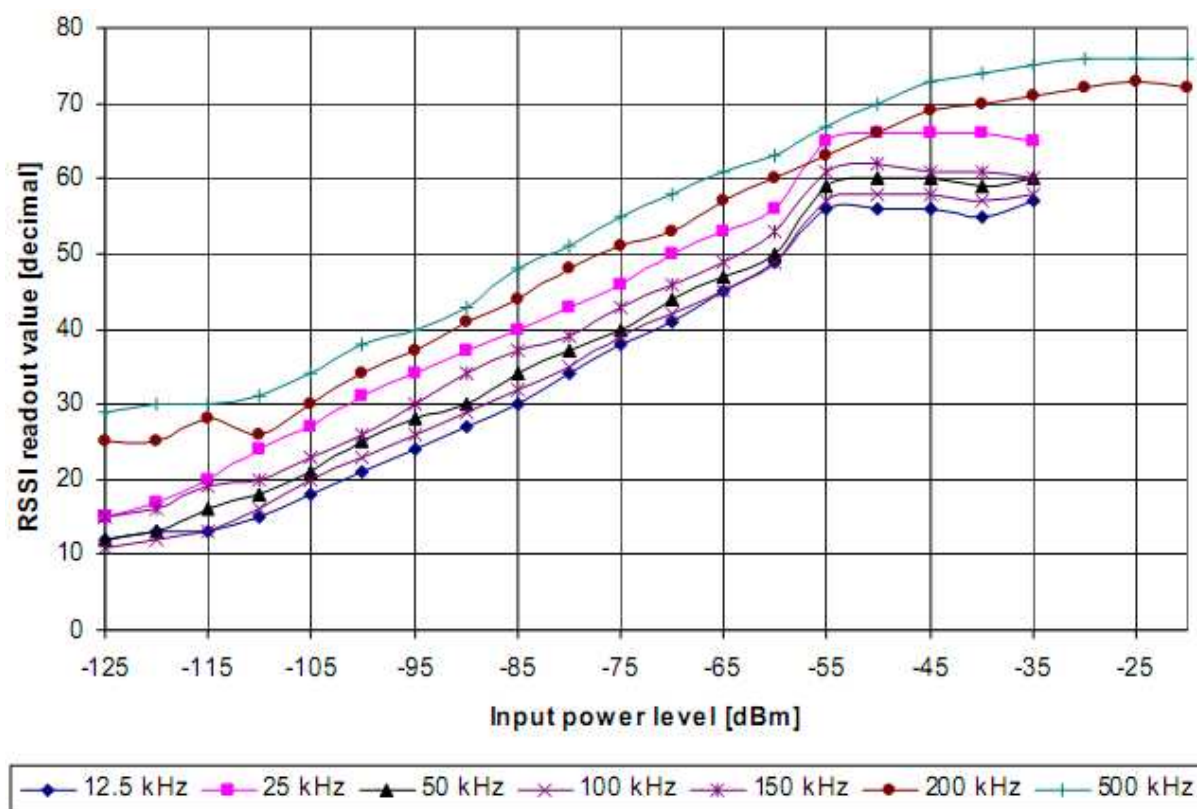
MYTNC <CMD><PAR>,

kde        CMD je příkaz, 2 znaky  
             PAR je parametr příkazu, proměnná délka.

Tab. 18 Seznam konfiguračních příkazů TNC

Příkaz	Parametr		Poznámka
	min	max	
TF	402000	407000	změna kmitočtu vysílače, jednotka kHz
RF	402000	470000	změna kmitočtu přijímače, jednotka kHz
TD	0	50000	změna kmitočtového zdvihu vysílače, jednotka Hz
RD	2000	50000	změna kmitočtového zdvihu přijímače, jednotka Hz
BR	1200	19200	změna modulační rychlosti, viz modulační rychlosti
PW	0	25	změna výkonu vysílače, hodnoty 0,3,6,9,12,15,18,21,25
RS	SI	-	dotaz na sílu přijímaného signálu

Příkaz „MYTNC TF435000“ provede změnu pracovního kmitočtu vysílače na 435 MHz. Příkazem „MYTNC RSSI“ se zjišťuje síla přijímaného signálu. TNC v okně zprávy odpoví: „RSSI = XXX“, kde XXX je trojčíselné vyjádření intenzity signálu. Závislost číselného vyjádření na intenzitě signálu ukazuje obr.28. Od hodnot intenzity je potřeba odečíst zisk vstupního zesilovače 20 dB. Šířka pásma kanálu se spočítá podle vztahu (9) s koeficientem 1,5.



Obr. 28 Typické hodnoty RSSI pro pásmo 433 MHz a různé šířky pásma kanálu

## 7 Závěr

Z úvodního nástinu struktury paketu protokolu AX25 a vlastností a funkce obvodu CC1020 je zřejmé, že obvod je možné provozovat v aplikacích daného typu. V režimu vysílání obvod nepřináší žádná omezení, v režimu příjmu je nutno dbát na minimální poměr datové rychlosti a modulačního zdvihu.

Data paketu AX25 jsou scamblována, čímž je účinně potlačena stejnosměrná složka signálu. Na začátku paketu se vysílá posloupnost hlaviček, není možné vysílat pouze jednu hlavičku, neboť na začátku příjmu dochází při descamblování ke ztrátě 17 bitů. Je zřejmé, že takto sestavený paket vyhoví potřebě obvodu nastavit filtr bitového komparátoru demodulátoru na začátku příjmu.

Navržený modul obsahuje obvod CC1020 firmy Chipcon (součást společnosti Texas Instruments), koncový PA stupeň se ziskem cca 15 dB, což umožňuje dosáhnout výstupního výkonu 300 mW, a vstupní LNA stupeň se ziskem cca 20 dB, což mělo umožnit dosáhnout vstupní citlivosti přibližně -135 dBm. Skutečná citlivost přijímače je -90 dBm, zjištěná citlivost u dvou vzorků obvodu CC1020 byla změřena na cca -70 dBm. Návrh zesilovačů je výsledkem simulací v prostředí *Ansoft Serenade 8.5*. V simulacích byl i zahrnut vliv přepínače příjem – vysílání a výstupního filtru vyšších harmonických složek. Modul je navržen na oboustrannou desku plošného spoje, součástí modulu není řídicí mikrokontrolér.

Softwarová část práce byla rozdělena do dvou modulů. Modul **cc1020.c** je ovladačem obvodu a obsahuje základní funkce pro nastavování řídicích registrů obvodu a přepínání režimů práce. Bez funkcí tohoto modulu není možné obvod provozovat. Modul **tnc.c** představuje řešení TNC dle zadání. Obsahuje algoritmy pro správu přijímaných a vysílaných dat, jejich kódování a dekodování.

Během pokusů a laboratorních měření se ukázalo, že domněnka vhodnosti obvodu, na základě parametrů tab. 1, pro packet rádio s rychlostí přenosu 9600 Bd byla mylná. Při snižování zdvihu modulace pod cca  $\pm 5000$  Hz prudce narůstá chybovost přijímaných data a při požadovaném zdvihu  $\pm 2750$  Hz je již příjem nemožný. Obvod tak pouze pro daný zdvih vyhoví do přenosové rychlosti 4800 Bd.

Výsledkem práce jsou dva funkční moduly pro aplikace rádiového přenosu dat s obvodem CC1020. Spolu s předkládaným softwarem pro mikrokontroléry Atmel AVR ATmega 128 a ATmega 16 a terminálovým programem AGW tvoří funkční TNC.

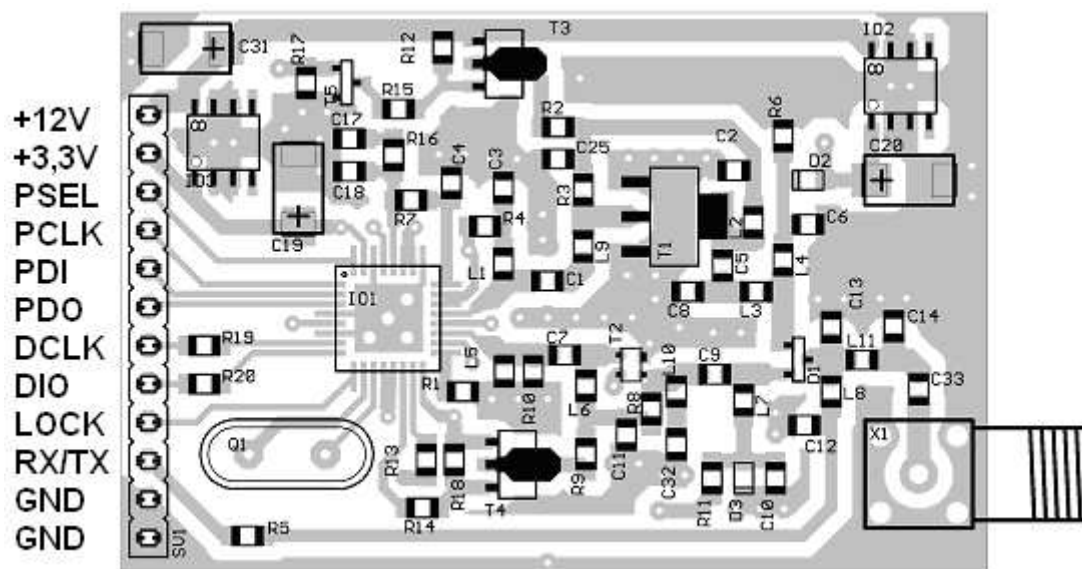
## 8 Seznam použité literatury

- [1] HLAVICA, P. *Realizace datového transceiveru pro packet rádio s obvodem fy. Chipcon včetně TNC v KISS módu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2006. 30 s. Vedoucí bakalářské práce Ing. Tomáš Urbanec, Ph.D.
- [2] HLAVICA, P. *Transceiver a TNC pro datovou komunikaci na UHF s obvodem CC1010*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2006. 7 s. Vedoucí semestrálního projektu Ing. Tomáš Urbanec, Ph.D.
- [3] *CC1020 Single Chip Low Power RF Transceiver for Narrowband Systems* [online]. Chipcon, Texas Instruments, 2007, URL <<http://www.ti.com/lit/gpn/cc1020>>, [cit. 2007-12-03]
- [4] *AN023 -- CC1020 MCU Interfacing* [online]. Chipcon, Texas Instruments, 2007, URL <<http://www.ti.com/litv/pdf/swra069>>, zdrojový soubor programu <<http://www.ti.com/litv/zip/swra069>>, [cit. 2007-12-03]
- [5] *AX.25 Amateur Packet-Radio Link-Layer Protocol, Version 2.0, October 1984* [online]. URL <[http://www.tapr.org/pub\\_ax25.html](http://www.tapr.org/pub_ax25.html)> [cit. 2007-10-01]
- [6] *Popis protokolu AX.25* [online]. URL <<http://www.qsl.net/ok0ns/AX25/AX25.TXT>>, [cit. 2007-10-01]
- [7] CHEPPONIS, M., KARN, P. *The KISS TNC: A simple Host-to-TNC communications protocol*. [online]. 1997. URL <<http://www.ka9q.net/papers/kiss.html>> [cit. 2007-10-01].
- [8] SCHIEFER, J. *KISS and SMACK Protocol for communication via the TNC* [online]. URL <http://www.symek.com/g/smack.html> [cit. 2007-10-01]
- [9] ROSSOPOULOS, G. *AGW Packet Engine* [online]. Ver 2004.1108. Počítačový program pro packet rádio, volně stažitelný, kompatibilní s Win95/98/NT/2k/ME/XP. URL <<http://www.sv2agw.com/downloads/agwpe.zip>>
- [10] ROSSOPOULOS, G. *AGWterm* [online]. Ver 95.15, 1998-09-09. Počítačový program pro packet rádio, volně stažitelný, kompatibilní s Win95/98/NT/2k/ME/XP. URL <<http://www.sv2agw.com/downloads/agwterm.zip>>
- [11] MILLER, J. *9600 Baud Packet Radio Modem Design* [online]. 2005-08-31. URL <<http://www.amsat.org/amsat/articles/g3ruh/109.html>> [cit. 2007-10-01]
- [12] FRAJLACH, K. *Paket rádio*. PVT reprog. sekce České Budějovice 1994, ISBN 80-900046-3-6

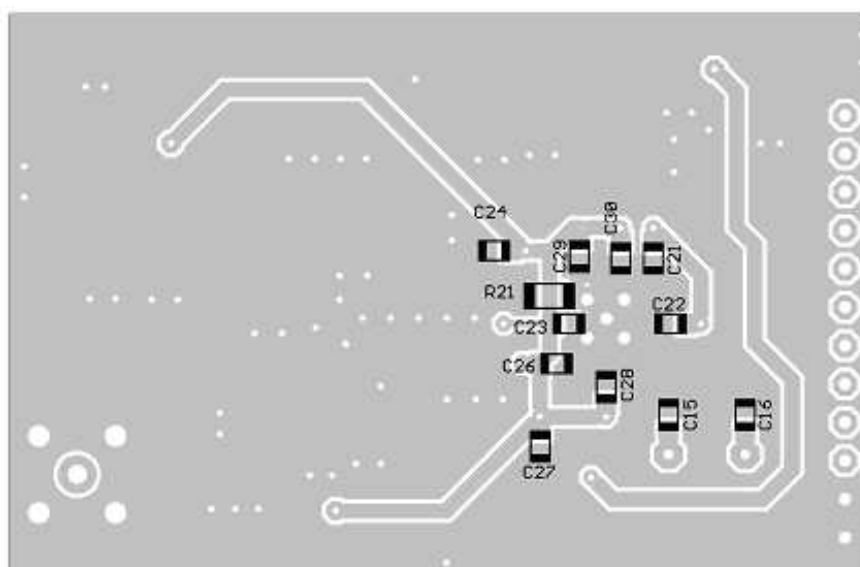


## 9 Přílohy

### 9.1 Deska plošného spoje



Obr. 29 Osazení modulu ze strany součástek a popis pinů



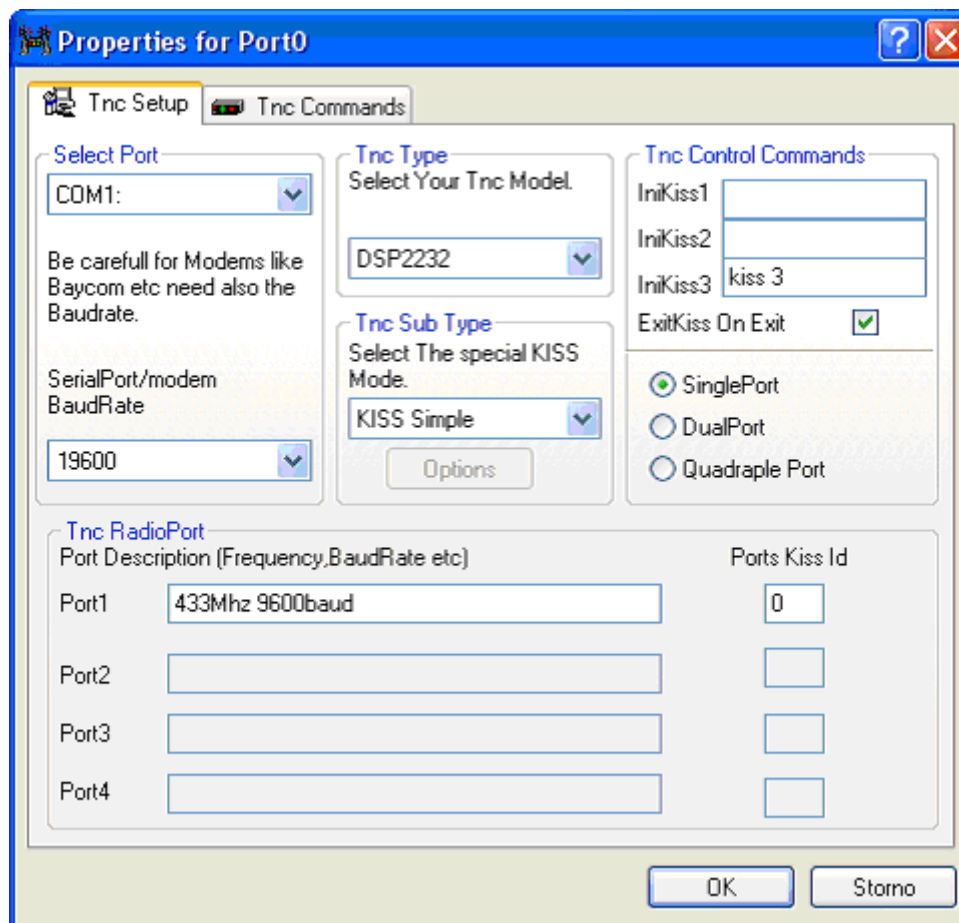
Obr. 30 Osazení modulu ze strany spojů

## 9.2 Seznam součástek

Součástka	Hodnota	Pouzdro			
			L1	8n2	R0805
			L2	100n	R0805
C1	15p	C0805	L3	39n	R0805
C2	100p	C0805	L4	100n	R0805
C3	100p	C0805	L5	27n	R0805
C4	560p	C0805	L6	220n	R0805
C5	100p	C0805	L7	100n	R0805
C6	100p	C0805	L8	100n	R0805
C7	100p	C0805	L9	3n3	R0805
C8	1p8	C0805	L10	27n	R0805
C9	2p7	C0805	L11	15n	R0805
C10	100p	C0805	Q1	14.7456MHz	HC49/U3
C11	100p	C0805	R1	82k	R0805
C12	100p	C0805	R2	22R	R0805
C13	15p	C0805	R3	15k	R0805
C14	15p	C0805	R4	82R	R0805
C15	12p	C0805	R5	1k	R0805
C16	12p	C0805	R6	10k	R0805
C17	56n	C0805	R7	10k	R0805
C18	2n2	C0805	R8	15k	R0805
C19	100u	CT6032	R9	220R	R0805
C20	100u	CT6032	R10	56R	R0805
C21	100p	C0805	R11	10k	R0805
C22	100p	C0805	R12	10k	R0805
C23	100p	C0805	R13	2k7	R0805
C24	100p	C0805	R14	2k7	R0805
C25	100p	C0805	R15	1k	R0805
C26	100p	C0805	R16	3k3	R0805
C27	100p	C0805	R17	22k	R0805
C28	100p	C0805	R18	22k	R0805
C29	100p	C0805	R19	1k	R0805
C30	100p	C0805	R20	1k	R0805
C31	10u	CT6032	R21	0R	R1206
C32	100p	C0805	T1	BFG591	SOT-223
C33	100p	C0805	T2	BFP540	SOT-343
D1	HSMP-3822	SOT-23	T3	BCX52	SOT89
D2	I.02	R0805	T4	BCX52	SOT89
D3	I.02	R0805	T5	BC817-40	SOT-23
IO1	CC1020	QFN32	SV1	PRSL20	
IO2	LE33	SO-8	X1	RF konektor	
IO3	LE33	SO-8			

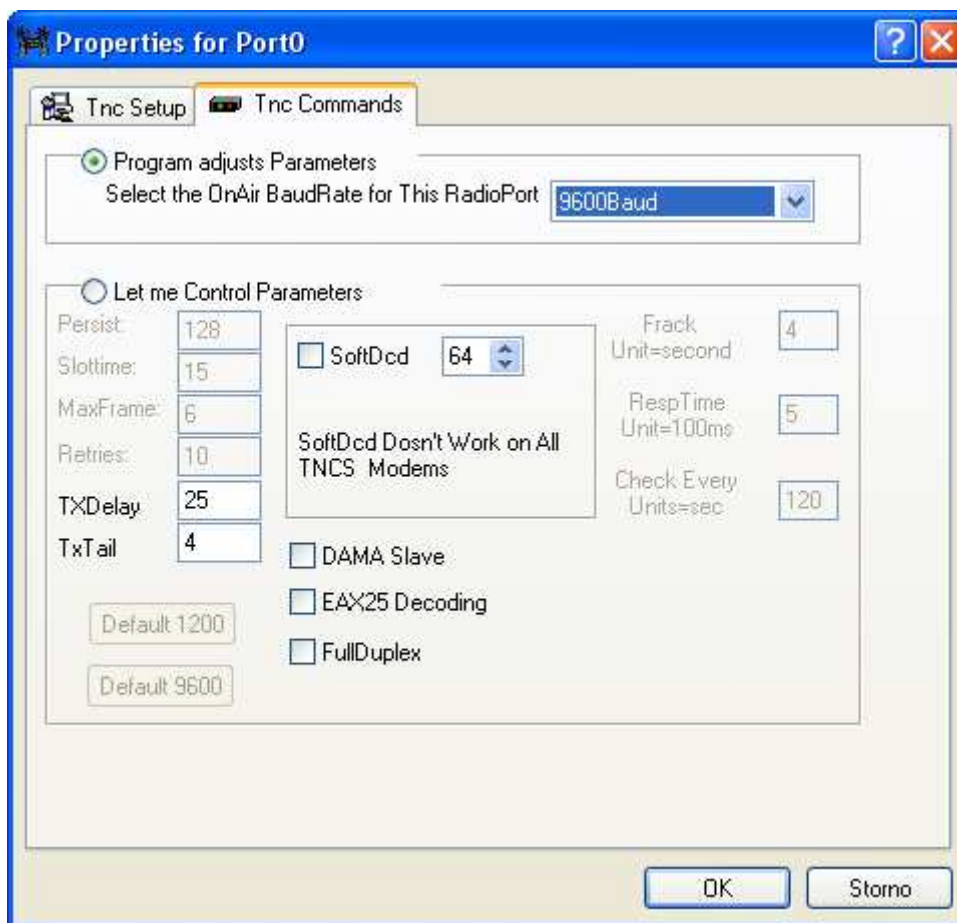
### 9.3 Instalace a nastavení terminálového programu

Z přiloženého CD se zkopíruje adresář AGW kamkoliv na pevný disk osobního počítače. Spustím se program *AGW Packet Engine*, tento program neběží jako okno, ale pouze jako tray ikona. Jako první krok je potřeba nakonfigurovat rádiový kanál. Kliknutím myši na tray ikonu se zobrazí menu, volím se položku *Properties*. Zobrazí se okno *PadioPort Selection*. Volím se *New Port* a v následujícím okně *AGWTNC* se potvrdí tlačítkem *OK*. Zobrazí se okno *Properties for Port0*, obr. 33.



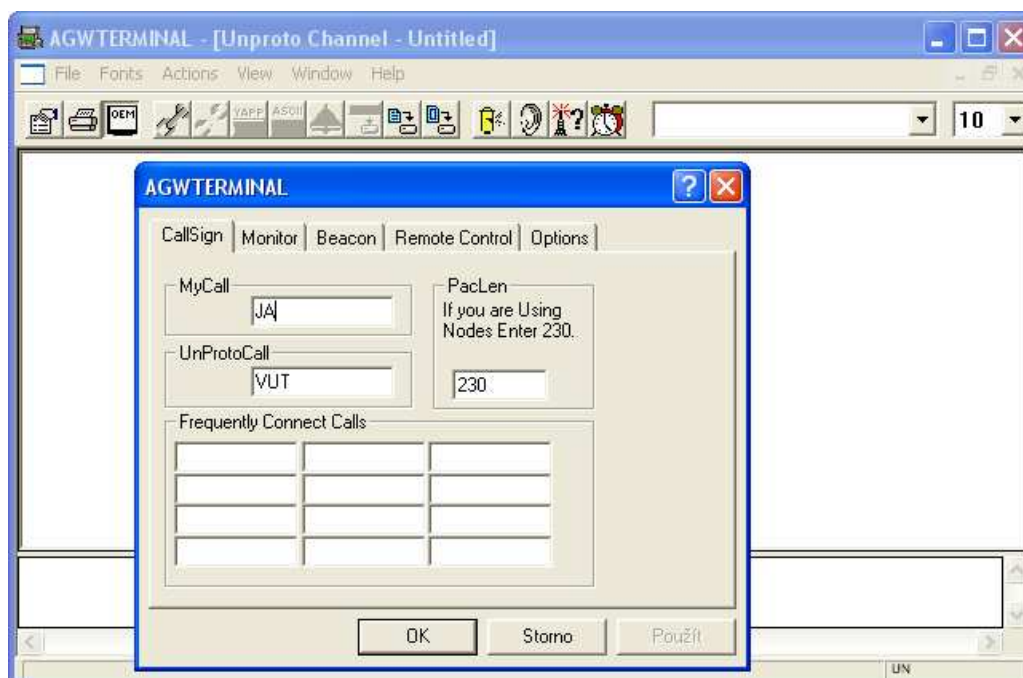
Obr. 31 Konfigurace rádiového kanálu - TNC Setup

V poli *Select Port* se nastaví neobsazený sériový port počítače, rychlost 19200 Baud. V poli *TNC RadioPort* se může konfigurovaný kanál pojmenovat, toto má význam pouze při používání více kanálů. Další pole se nastaví podle obrázku. Parametry na záložce *TNC Commands*, obr. 34, se nastaví dle obrázku a program se restartuje.



Obr. 32 Konfigurace rádiového kanálu - TNC Commands

Je-li již spuštěný a nastavený *AGW Packet Engine*, spustí se program *AGWterm*, obr. 35. Nyní je nutné nastavit volací značky, to se provede volbou *File/Properties*.



Obr. 33 Terminálový program AGWterm

Zobrazí se okno *AGWTERMINAL* poli *MyCall* se nastaví vlastní volací značka, v poli *UnProtoCall* se nastavím volací značka protistanice. Potvrdí se tlačítkem *OK*, změna vlastní volací značky vyžaduje restart programu *AGWterm*.